

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИПЕНКО
(підпис)

“ ____ ” _____ 20__ р

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: «Застосунок для миттєвого оповіщення студентів»

Виконав:

студент IV курсу, групи Ю-64

_____ Ткаченко Павло Олександрович _____

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

_____ ст. викладач Габінет Артем Вікторович _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Консультант

_____ н. контроль _____

(назва розділу)

_____ доц. д.т.н. Сімоненко В.П. _____

(вчені ступінь та звання, прізвище, ініціали)

_____ (підпис)

Рецензент

_____ доц. каф. СПіСКС к.т.н., доц Марія ОРЛОВА _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМ. ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИПЕНКО

(підпис)

“ ____ ” _____ 20__ р.

ЗАВДАННЯ

на дипломний проект студенту

Ткаченку Павлу Олександровичу

1. Тема проекту «Застосунок для миттєвого оповіщення студентів»

керівник проекту Габінет Артем Вікторович, старший викладач, кандидат
технічних наук, затверджені наказом по університету від « ____ » _____
_____ 2020р. № _____

2. Термін здачі студентом закінченої роботи _____ 2020р.

3. Вихідні дані до проекту технічне завдання, теоретичні дані.

4. Зміст пояснювальної записки: опис можливостей та функціоналу телеграм-ботів, їх створення та вдосконалення, застосунок для миттєвого оповіщення студентів.

5. Консультант роботи, з вказівкою розділів роботи, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В.П.		

6. Дата видачі завдання 01.09.2019 року

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1.	<i>Затвердження теми роботи</i>	01.09.2019	
2.	<i>Вивчення та аналіз завдання</i>	15.09.2019	
3.	<i>Розробка архітектури та загальної структури телеграм-бота</i>	04.10.2019	
4.	<i>Розробка структур окремих функцій бота</i>	13.12.2019	
5.	<i>Програмна реалізація бота</i>	03.02.2020	
6.	<i>Оформлення пояснювальної записки</i>	04.04.2020	
7.	<i>Передзахист</i>	26.05.2020	
8.	<i>Захист</i>	25.06.2020	

Студент

Павло ТКАЧЕНКО

(підпис)

Керівник

Артем ГАБІНЕТ

(підпис)

Анотація

В цьому бакалаврському дипломному проекті був розроблений телеграм-бот для миттєвого оповіщення студентів на мові програмування Python.

Пропонуємий функціонал даного телеграм-боту призначений для більш якісного та швидкого обміну інформацією між студентами, викладачами та адміністрацією КПІ.

Annotation

In this bachelor's thesis project, a telegram bot was developed to instantly alert students in the Python programming language.

The proposed functionality of this telegram-bot is designed for better and faster exchange of information between students, teachers and the KPI administration.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проект	2	
2	A4	ІАЛЦ.467200.001 ВП	Відомість проекту	1	
3	A4	ІАЛЦ.467200.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.467200.003 ПЗ	Пояснювальна записка	61	
5	A4	ІАЛЦ.467200.004 Д1	Принципова схема апаратного забезпечення телеграм-боту	1	
6	A4	ІАЛЦ.467200.005 Д2	Структурна схема телеграм-боту	1	
7	A4	ІАЛЦ.467200.006 Д3	Функціональна схема телеграм-боту	1	

					<i>ІАЛЦ.467200.001 ВП</i>			
Зм.		№ документа	Підпис	Дата	Відомість дипломного проекту	Літ.	Аркуш	Аркушів
Розробив	Ткаченко П.О.						1	1
Перевірів	Габінет А.В.					НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ гр. ІО-64		
Н. Контр.	Сімоненко В.П.							
Затвердив	Стіренко С.Г.							

ТЕХНІЧНЕ ЗАВДАННЯ

**до дипломної роботи
освітньо-кваліфікаційного рівня бакалавр**

на тему: “Застосунок для миттєвого оповіщення студентів”

Київ – 2020 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до розробленого продукту	3
5.2. Вимоги для портативних комп'ютерів	3
5.3. Вимоги до мобільних пристроїв.....	3
6. ЕТАПИ РОЗРОБКИ	4

					<i>ІАЛЦ.467200.002 ТЗ</i>		
Зм.		№ документа	Підпис	Дата	Застосунок для миттєвого оповіщення студентів Технічне завдання		
Розробив		Ткаченко П.О.					
Перевішив		Габінет А.В.					
Н. Контр.		Сімоненко В.П.					
Затвердив		Стіренко С.Г.			<div>Літ.</div> <div>Аркуш</div> <div>Аркушів</div> <div>1</div> <div>4</div> <div>НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ гр. ІО-64</div>		

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Даше технічне завдання також поширюється на курс «Оптимізація інформації у комп'ютерних системах». Область використання: практичне використання студентами та викладачами в освітньому процесі університету.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи кваліфіційно-освітнього рівня «бакалавр комп'ютерної інженерії», затверджене кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського» та покращення обміну інформації в університеті.

3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є створення телеграм-боту, який буде надсилати потрібну інформацію про навчання в університеті швидко та якісно, використовуючи мобільний додаток Telegram.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є електронна науково-технічна література про Telegram API, публікації в Інтернеті, різні статті про телеграм-ботів.

						Лист
						2
Зм.	Арк.	№ документа	Підпис	Дата		

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробляемого продукту

- Телеграм-бот може опрацьовувати декілька користувачів одразу.
- Телеграм-бот не буде відповідати на некоректні запити.

5.2. Вимоги для портативних комп'ютерів

- ОС: Windows XP, 7, Vista, 8, 8.1, 10.
- Процесор: Intel Celeron 1800 MHz.
- Оперативна пам'ять: 256 MB ОЗУ.
- Відеокарта: Intel HD Graphics.
- Місце на диску: 128 MB.

5.3. Вимоги до мобільних пристроїв

- Android 2.2 або вище.
- IOS 6.0 або вище.

					ІАЛЦ.467200.002 ТЗ	Лист
Зм.	Арк.	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	28.03.2020
Складання і узгодження технічного завдання	03.04.2020
Створення модулів системи, що розробляється	15.04.2020
Тестування окремих модулів бота	25.04.2020
Допрацювання, налагодження і виправлення помилок	01.05.2020
Оформлення документації дипломної роботи	15.05.2020

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту
на тему: «Застосунок для миттєвого оповіщення студентів»

Київ – 2020 року

ЗМІСТ

ЗМІСТ.....	1
СПИСОК СКОРОЧЕНЬ	2
ВСТУП	3
ПОСТАНОВКА ЗАДАЧІ	4
РОЗДІЛ 1. TELEGRAM-БОТ — ЯК ОДИН З ШЛЯХІВ РЕАЛІЗАЦІЇ ДАНОГО ПРОЕКТА	5
1.1 Telegram	5
1.2 Боти в Telegram	5
1.3 Переваги ботів телеграму	7
1.4 Види чат-ботів.....	9
1.5 Приклади різних існуючих ботів	11
ВИСНОВОК ДО РОЗДІЛУ 1	19
РОЗДІЛ 2. ВИБІР ЗАСОБІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ	20
2.1 Розробка на JavaScript	20
2.2 Веб-система Django	22
2.3 Система управління БД MongoDB та MySQL.....	24
2.4 Шляхи оновлення бота: Webhook та getUpdates	29
2.5 Розробка на Python	31
ВИСНОВОК ДО РОЗДІЛУ 2	35
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ДИПЛОМНОГО ПРОЕКТУ НА МОВІ ПРОГРАМУВАННЯ PYTHON	36
3.1 Мова програмування Python	36
3.2 Середовище розробки IDE PyCharm	42
ВИСНОВОК ДО РОЗДІЛУ 3	43
РОЗДІЛ 4. ОПИС ВИКОРИСТАНИХ ФУНКЦІЙ ТА ФУНКЦІОНАЛЬНІСТЬ ТЕЛЕГРАМ-БОТУ	44
4.1 Створення телеграм-боту. Опис використаних функцій.....	44
4.2 Застосунок для миттєвого оповіщення студентів	46
ВИСНОВОК ДО РОЗДІЛУ 4	55
ВИСНОВОК ДО ДИПЛОМНОГО ПРОЕКТУ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57

					ІАЛЦ.467200.003 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ткаченко П.О.			Застосунок для миттєвого оповіщення студентів Пояснювальна записка	Літ.	Арк.
Перевір.		Габінет А.В.					12
						НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ гр. ІО-64	
Н. Контр.		Сімоненко В.П.					
Затверд.		Стіренко С.Г.					
						Акрушів	59

СПИСОК СКОРОЧЕНЬ

API - Application Programming Interface

HTTP - HyperText Transfer Protocol

HTML - HyperText Markup Language

ORM - Object-Relational Mapping

БД – База Даних

СУРБД - Система Керування Базами Даних

ISAM - Indexed Sequential Access Method

SQL - Structured Query Language

GUI - Graphical User Interface

IDE - Integrated Development Environment

ООП – Об’єктно-Орієнтоване Програмування

					ІАЛЦ.467200.002 ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Метою даного проекту є спрощення отримання інформації студентами з кампусу та інших офіційних сайтів КПП. Цей проект має багато переваг. На сьогоднішній день не кожен студент завжди може отримувати інформацію від кампусу, або офіційних сайтів “КПП” своєчасно, деякі студенти не користуються взагалі, або дуже рідко.

Більшість студентів використовують у повсякденному житті месенджер Telegram, тому це стало однією з багатьох причин створення дипломного проекту на основі телеграм-бота.

Відкривши документацію по Telegram Bot API можна дізнатися, що це HTTP інтерфейс, який надає можливість розробникам створювати різних ботів в Telegram. Телеграм-боти мають багато можливостей та функцій, які інколи обмежуються лише уявою розробника.

Для успішного навчального процесу студентів, головне, це швидке отримання інформації та її завчасне оновлення.

У кожного з студентів є безліч шляхів встановлення зв'язку з викладачем та адміністрацією кафедри, але не всі вони зручні та ефективні. Тому метою цього дипломного проекту буде створення телеграм-бота, який буде надавати інформацію студентам швидко та зручно.

Розроблений телеграм-бот повинен надавати можливість швидкого спілкування та обміну інформацією між студентами та викладачами, або керівництвом кафедри.

					ІАПЦ.467200.003 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ПОСТАНОВКА ЗАДАЧІ

Отже метою цього проекту є створення спеціального чат-боту для зручного та швидкого оповіщення студентів. Отримання інформації не буде займати багато часу та буде систематизовано, студенти зможуть використовувати меню бота для пошуку конкретної інформації.

					ІАПЦ.467200.003 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. TELEGRAM – ЯК ОДИН З СПОСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

1.1 Telegram

Telegram на даний момент має великий функціонал. Один з них — Bot API. Чат боти вже дуже широко поширені, як серед розробників-початківців, так і у бізнесі. Чому можна навчити бота?

1.2 Боти в Telegram

Великий функціонал полягає в[1]:

- Отриманні спеціальних сповіщень та новин:

Бот може відсилати найновішу інформацію, як тільки її публікують різні джерела інформації.

- Інтеграції з іншими службами:

Бот має можливість комунікувати з різними службами, як приклад, боти: GitHub bot, Gmail Bot, GIF bot, IMDB Bot, Wiki Bot, інше.

- Отриманні платежів від користувачів Telegram:

Телеграм-боти можуть приймати платежі та товари починаючи з версії API Bot v.3.0. Для користування телеграм-ботом знадобиться Telegram v.4.0 або вище, щоб була можливість оплати товару або послуги[2].

Торгові надають можливість відсилати користувачу спеціально відформатовані повідомлення рахунків-фактур. Ці рахунки містять фотографію, опис товару, тощо, а також кнопку оплати. Натискання цієї кнопки відкриває користувачу спеціальний платіжний інтерфейс в Telegram. У цьому інтерфейсі користувач може отримати додаткову інформацію про товар, а саме: інформацію про доставку, номер телефону, адресу пошти та інше.

Після того, як користувач введе всю необхідну інформацію – телеграм-бот пропонує користувачу декілька варіантів доставки товарів на адресу

					ІАПЦ.467200.003 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача.

Після готовності клієнту надається можливість вести дані своєї кредитної карти або обрати зі збережених карт – і оплатити продукт. Потім телеграм-бот надсилає квитанційне повідомлення(рис. 1.1).

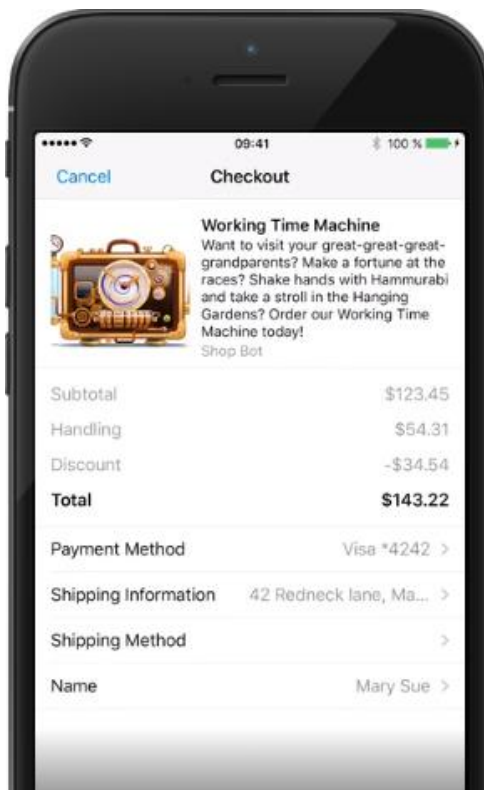


рис. 1.1 приклад квитанції від торгового боту.

Телеграм-боти підтримують також Apple Pay та Android Pay. Telegram не обробляє платіж, він перенаправляє їх на різних постачальників платежів по всьому світу. Самі обробники платежів займається транзакцією та зберігають конфіденційну інформацію. Ні сам Telegram, ні розробники ботів не мають доступу до обробки платежу. На даний час Telegram підтримує платежі більше ніж с 200 країн через наступних постачальників платежів:

- Stripe
- LiqPay
- Payme
- CLICK
- Tranzzo
- Sberbank

					ІАПЦ.467200.003 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

- Rave by Flutterwave
- Yandex.Money
- Створювання спеціальних інструментів:

Бот також вміє відправляти Вам оповіщення, прогнози погоди, переклади, різні форматування та інше. Як приклад: Markdown bot, Sticker bot, Vote bot, Like bot та інше

- Створення одиночних та багатокористувацьких ігор:

Telegram-боти мають багатий досвід в HTML5, а саме — від простих типів аркад та пазлів до 3D-стрілок та різних стратегічних ігор у режимі реального часу: GameBot. Gamee – як приклади.

- Будування соціальних послуг:

Бот може з'єднувати людей на основі їх спільних інтересів та хобі для спілкування.

- Виконанні будь-якої електронної роботи

1.3 Переваги ботів телеграму

Telegram-боти мають деякі особливості на відміну від звичайних користувачів:

- У ботів немає ніякого статусу <онлайн>, або <був недавно онлайн>, замість цього відображається мітка <бот>[3].
- Також боти надають користувачам обмежене по пам'яті хмарне зберігання – старіші повідомлення будуть видалені сервером після їх обробки.
- Боти не можуть почати чат з користувачем. Користувач повинен або додати їх до групи, або надіслати їм повідомлення. Також користувачі можуть скористатися t.me/<bot_username> посиланням, або пошуковим іменем, щоб знайти свого бота.
- Ім'я кожного телеграм-бота завжди закінчується на <bot>. Наприклад: @Trivial_bot, @Github_bot та інші.
- Якщо додати бота до групи – він не буде отримувати усі повідомлення за замовчуванням.

					ІАПЦ.467200.003 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- Боти працюють завжди і можуть відповідати користувачу у будь-якій годині дня і ночі.

Telegram-боти мають декілька незаперечних переваг над мобільними додатками, сайтами і групами в соц. Мережах[4]:

- Персональна увага до кожного користувача.

Кожен користувач будь-якого продукту обожає персональний підхід до себе і готовий платити за це. В світі є мільйони безликих сайтів та додатків на кожному з яких приходить орієнтуватися. Боти мають можливість працювати над конкретним питанням користувача окремо від інших й швидше переходити к його рішенням.

- Текстовий інтерфейс споживає менше трафіку.

Не потрібно чекати завантаження сайту, переходити з однієї сторінки на іншу, а це значить більш швидкий процес досягнення мети.

- Низька вартість

Telegram-боти можуть працювати навіть на звичайному комп'ютері, їм не обов'язково орендувати сервера як сайтам. Боти не потребують певну версію Android або iPhone, їм потрібен лише Telegram на мобільному телефоні або комп'ютері, який є безкоштовним для всіх.

- Гнучкість и швидкість відповіді

Telegram-боти відповідають одразу на команди за запитання, якщо вони на це запрограмовані, легкість переробки та зміни бота дозволяю швидше повернути його до роботи.

- Захист від конкурентів

Завдяки тому, що бот веде чат с клієнтом один-на-один, а не на відкритому інтернет просторі — це дозволяє в декій мірі захиститись від посягані конкурентів на власного клієнта.

- Не потребує установки та авторизації

Клієнту не потрібно реєструватися або проходити авторизацію, він починає працювати з ботом с першого повідомлення або команди.

- Зниження людського фактора

					ІАПЦ.467200.003 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Telegram-бот не забуде зв'язатися з користувачем, моментально відповість на запит та надасть всю потрібну інформацію користувачеві.

- Автоматизація виконання шаблонних справ.

Telegram-боти виконують усю рутинну працю набагато швидше та без помилок, наприклад: складання звітів, зведень, будь-яких актів, генерування документів та їх зберігання.

1.4 Види чат-ботів

Створення чат-боту обмежується тільки уявою та навичками програміста, але можна виділити деякі певні види чат-ботів[5]:

- Чат-бот для сфери послуг.
 - Отримання інформації про робочий графік, вихідні, вільні місця, тощо
 - Записи на прийом, співбесіди, тощо
 - Отримання зворотного зв'язку
 - Відправка повідомлень
 - Прийом оплати
 - Інтеграція з програмами лояльності
- Чат-бот для фінансових установ, банків[6].
 - Отриманні інформації о найближчем відділенні банкомату або банку
 - Інформування о послугах
 - Отримання заявок від користувача
 - Отримання рішення на популярні запитання
 - З'єднання з оператором у разі необхідності
 - Погашення кредиту, заборгованості, інтернет банкінг.
 - Інтеграція з системами банку.

					ІАПЦ.467200.003 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Чат-бот для служби підтримки.
 - База популярних запитань
 - Отримання, обробка та фіксування заявки, звернення, тощо
 - Додавання оператору в діалог у разі необхідності
 - Відслідкування обробки та статусу заявки користувача
- Чат-бот для служби доставки.
 - Повне меню закладу, оформлення замовлення
 - Оплата замовлення прямо в боті
 - Відправка акцій і спец-пропозицій клієнтам
 - Сегментація клієнтів
- Чат-бот для роботи з рекрутинговими агентствами, персоналом.
 - Первинне анкетування кадрів
 - Роботизація HR-процесів
 - Швидке надавання інформації персоналу або кадрам
 - Повна інформація потрібна персоналу
- Чат-бот для конференцій і форумів.
 - Інформація про план, графіки початку та кінця заходів
 - Інформація для про спікерів
 - Чати для спілкування гостей
 - Проведення опитування і тестів
 - Розсилка інформації та потрібних файлів під час заходів
- Чат-бот для бронювання та купівлі квитків.
 - Список вільних місць, номерів, інше
 - Оплата в боті
 - Підключення оператора
 - Отримання зворотного зв'язку
 - Розповсюдження акцій та спец. Пропозиції
- Чат-бот для туристичних агентств.
 - Підбір туру

- Налаштування параметрів бота для отримання цікавих для клієнта турів
- Розповсюдження акцій
- Зв'язок під час відпочинку
- Отримання відгуків та оцінок про тур
- Індивідуальні чат-боти.

У кожного є можливість розробити власного бота с будь-яким функціоналом адаптованим під завдання певної людини. Форматування, зберігання, розсилка, пошук, моніторинг, інше.

1.5 Приклади різних існуючих ботів

З появою чат-ботів в бізнесі відбулося багато змін. Компаніям та клієнтам цікавий цей інструмент. Багато брендів впроваджують телеграм-ботів в свій бізнес, спрощуючи собі обробку інформації, зберігаючи час для особливих непоширених проблем та операцій[7].

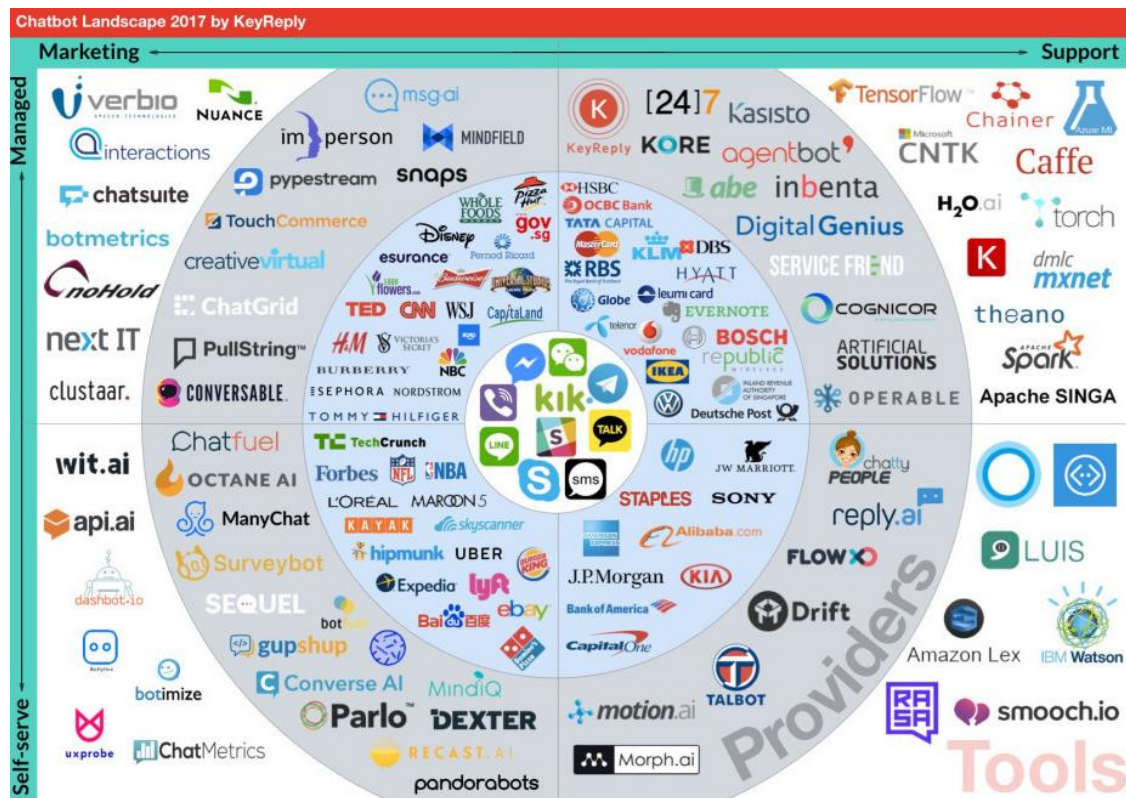


рис. 1.2 ландшафт індустрії використання чат-ботів, створений американським сервісом KeyReply.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

Опис сфери використання телеграм-ботів(рис. 1.2):

- Компанії розділені по осях залежно від їх різноманітних функцій, які виконують їх телеграм-боти (підтримка або маркетинг), і від того місця на рисунку, де вони створені.
- Коло в центрі – це месенджери, які користуються популярністю серед людей та дозволяють створювати ботів через різні API-інтерфейси.
- Наступне коло - Бренди – це компанії, які запустили і протестили ботів спеціального типу (наприклад, Managed x Support - бот для обробки замовлень в студії).
- Провайдери - компанії, які допомагають у створенні телеграм-ботів, а Tools - інструменти, якими користуються провайдери, бренди або розробник.

Популярні платформи для створення ботів:

- Facebook Messenger – чат-бот-платформа, яка має аудиторію розміром понад 1,2 мільярда постійних користувачів на місяць і більше 100 тисяч працюючих ботів в світі. Найбільш розповсюджена платформа серед месенджерів в США, а по кількості функцій - сама просунута для розробників: великий вибір елементів взаємодії (шаблони списку товарів, розширення, чеки, шаблони реєстрації на рейс), аутентифікація, прийом платежів.
- ВКонтакте - має API для ботів під спілкування від імені певної кількості людей зі своїми користувачами. Можна відправляти клієнтам як текст так й фото, файли, місце розташування, стікери. Але не має елементів різної взаємодії, які є в інших чат-бот платформах: кнопок, пунктів меню, розширень, шаблонів повідомлень.
- Viber - публік-акаунти, які є для створення бізнесу в Viber надають можливість компаніям налаштувати діалогове спілкування з передплатниками свого акаунту - через бота або CRM. Серед інсуючих чат-ботів на платформі чат-бот: журнал Glamour, App in The Air, Aviasales.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Telegram - одним з перших надав можливість створювати ботів в месенджері. За даними Forbes, в Telegram існує десь 100 тисяч спамерських пошукових ботів. Російські боти: чат-бот «Ощадбанку», OneTwoTrip_Bot, або наприклад бот «Пошти Росії», видання «Медуза» і так далі. Не так давно чат-боти в Telegram отримали можливість приймати платежі від користувачів, в Росії прийом платежів здійснює Яндекс.Каса.
- WhatsApp - у WhatsApp на даний час також є офіційні акаунти для бізнесу. Відомо, що WhatsApp тестує API з низкою компаній на ринках Індії, Бразилії та Пакистану - наприклад, фермери можуть відслідковувати повідомлення про фізичний стан своїх корів та іншу корисну інформацію, яку потрібно оновлювати завжди.

Використання ботів брендами.

Маркетингові боти зазвичай використовують для коротких маркетингових функцій. Ефективність вимірюється кількістю отриманих користувачів, конверсією в продажі і прибутком.

Боти підтримки існують та працюють набагато довше, і клієнти вже звикли до них. Тут використовуються такі параметри, як швидкість відповідей, задоволеність користувачів, зниження числа дзвінків в колл-центр і так далі.

Приклади існуючих ботів великих компаній:

- Бот каршерінга YouDrive

Платформа: Telegram.

Бот на 2/3 знизив кількість дзвінків клієнтів в компанію.

Бот YouDrive розроблювався штатними розробниками компанії. Сценарії та функції створювалися після аналізу певних типів звернень клієнтів через інші канали комунікації компанії - телефон, пошту і соц. мережі.

Мета бота - зробити процес ознайомлення з проблемами з автомобілем або орендою простим і зручним для клієнта, а також підвищити швидкість обробки і вирішення проблеми, знизивши для клієнта час очікування й обробки рішення.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

По-перше, бот пропонує готові відповіді на різні питання: від заведення автомобілю до «що робити, якщо я потрапив в ДТП».

По-друге, - розподіляє кількість різних звернень від користувачів по відповідальним співробітникам: кожен тип звернення відправляється до певного відділу для оперативного вирішення проблеми.

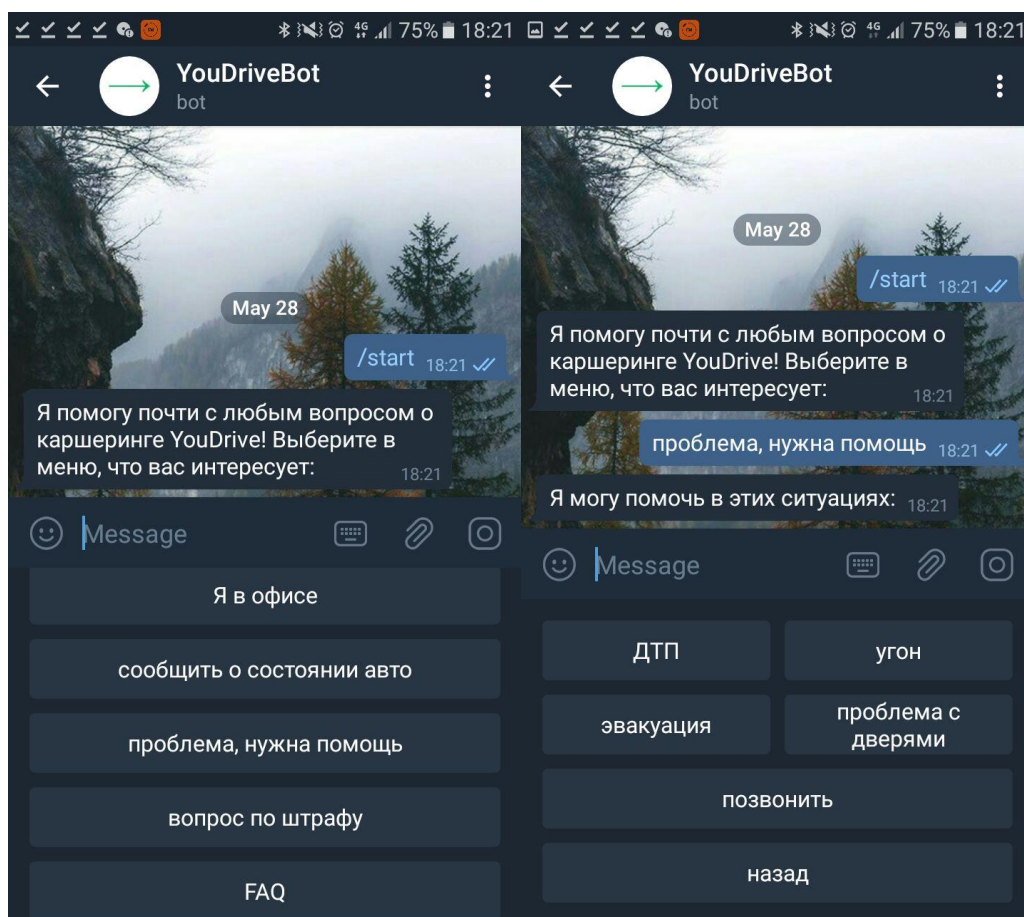


рис. 1.3 вигляд та функціонал боту @YouDriveBot

- Бот Rextester[8].

Один з цікавих представників телеграм-ботів є Rextester.

Функція бота – компіляція та виконання певних фрагментів коду на мові, яку обере користувач зі зазначеного списку мов програмування. Корисний бот, якщо в розробника раптом немає комп'ютеру під рукою, а певний код треба протестувати(рис 1.4).

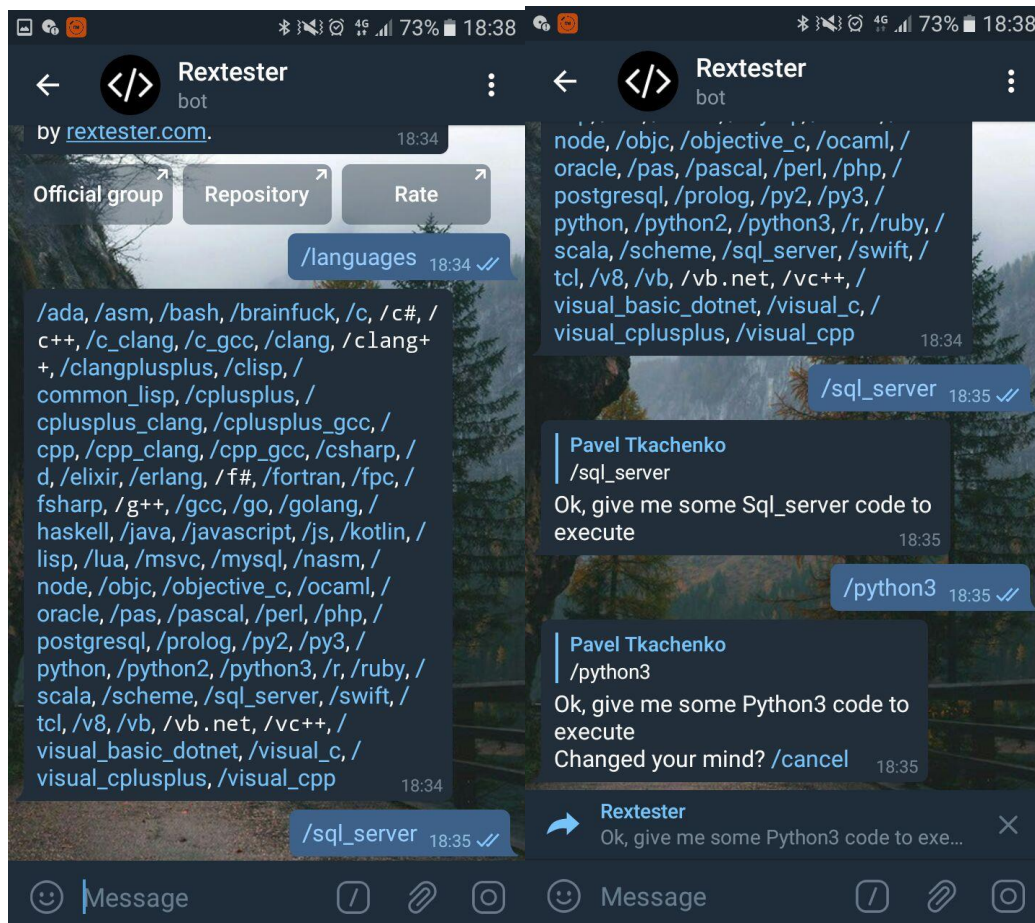


рис 1.4 інтерфейс та функціонал боту Rextester

- Бот @AviasalesTicketBot[9].

Цей бот відслідковує ціну квитків по заданих напрямках користувачем. Як тільки ціна квитка отримує нижче значення - бот надсилає повідомлення користувачеві про цю зміну. Користувачу потрібно тільки обирати напрямки, дати і розмір знижки, при якій бот надішле повідомлення.

Для управління цим телеграм-ботом має бути присутня підписка.

Як тільки користувач отримає підписку він зможе вказати аеропорт, місто або країну відправки і кінцеву країну. Користувач може обрати функцію «Куди завгодно» в пункті призначення, якщо користувачу просто цікаві низькі ціни, але байдужі у які летіти напрямки. Тоді бот буде відслідковувати за цінами усіх можливих рейсів, що вилітають із зазначеного міста. Знижка на переліт – один з особливих параметрів. Клієнт може встановити знижку від 30%, та буде отримаувати повідомлення, коли ціна на квиток зміниться, на

					ІАПЦ.467200.003 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

30% нижче від середньої ціни перельоту у цьому напрямку(рис. 1.5).

Чим менше зазначена знижка користувачем, тим більша буде кількість квитків. Якщо користувач отримує забагато повідомлень про знижки то йому слід лише збільшити відсоок знижки для зменшення повідомлень.

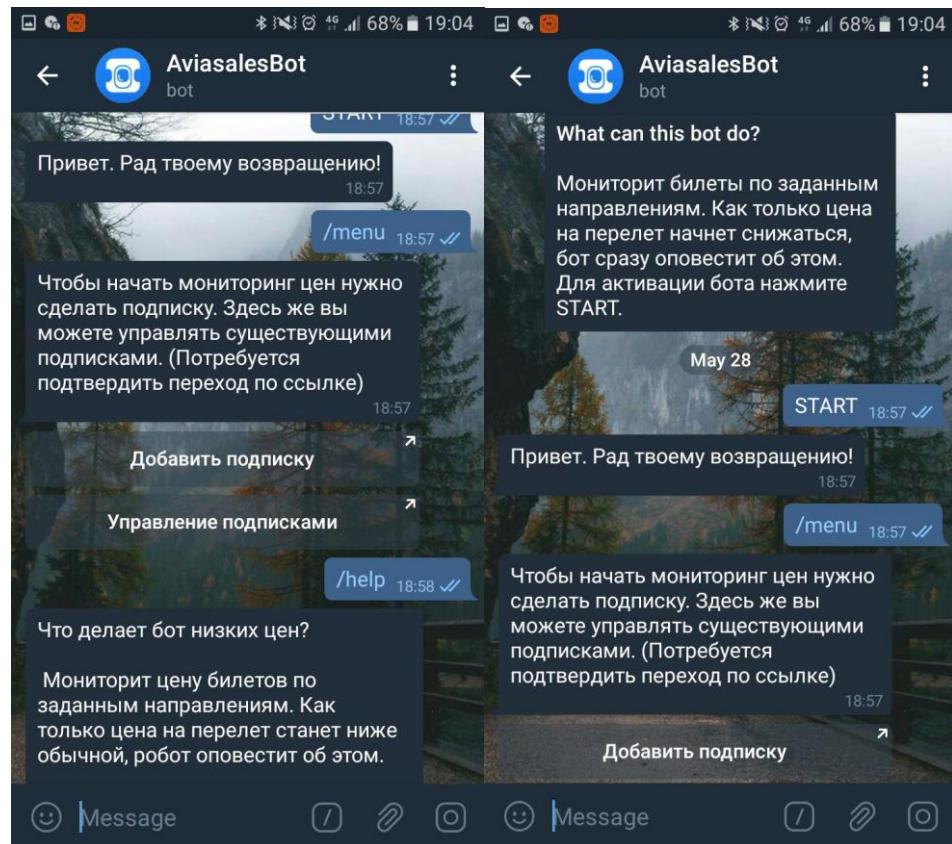


рис. 1.5 інтерфейс боту @AviasalesTicketBot

- Бот @YouTubeAudioDownloadChat.

Основна задача цього боту – надсилання звукової доріжки з відео(рис. 1.6), яке знаходиться на YouTube. Також клієнт може написати боту назву пісні й бот запропонує клієнту декілька відео, з яких він може обрати варіацію по його смаку.

- @LingvoBot.

Основна функція цього телеграм-боту це перекладання з англійської на російську та навпаки. Видає також коротке тлумачення слів. Непоганий бот для тих, хто вивчає англійську(рис. 1.7).

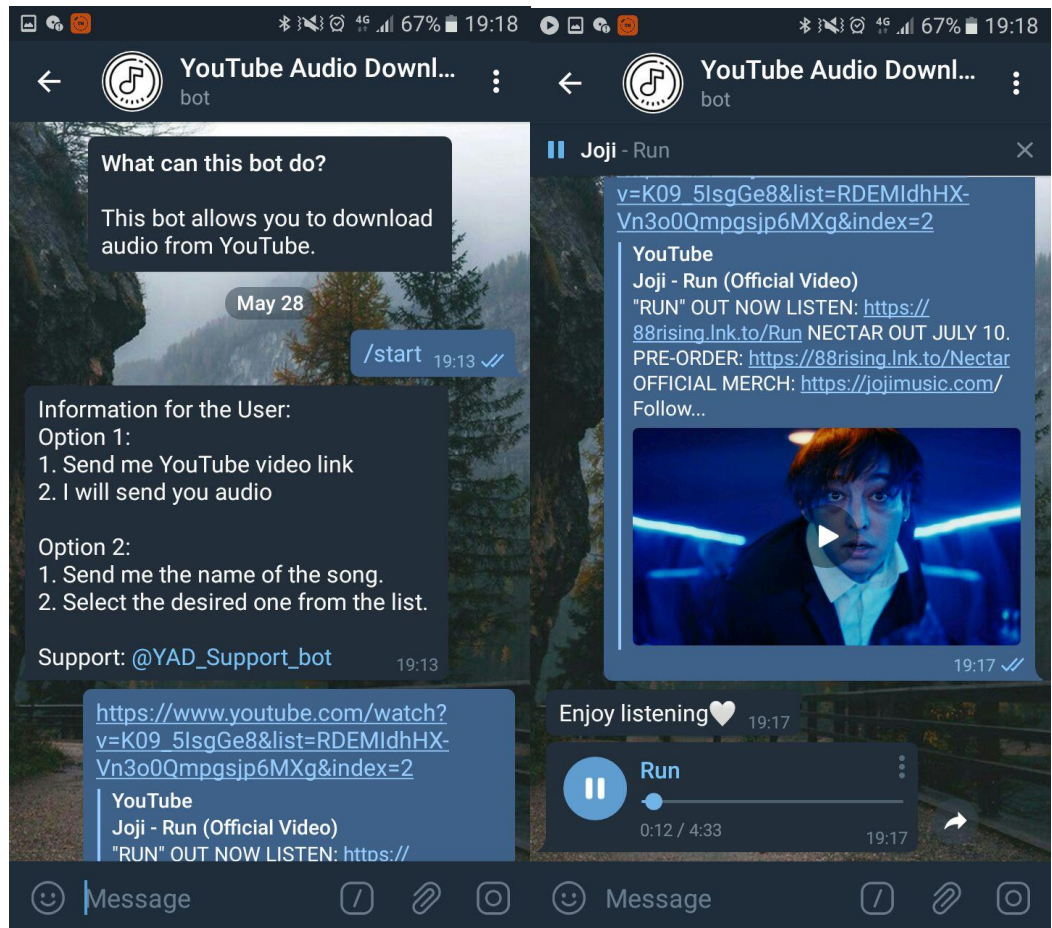


рис. 1.6 приклад дії боту @YouTubeAudioDownloadChat.

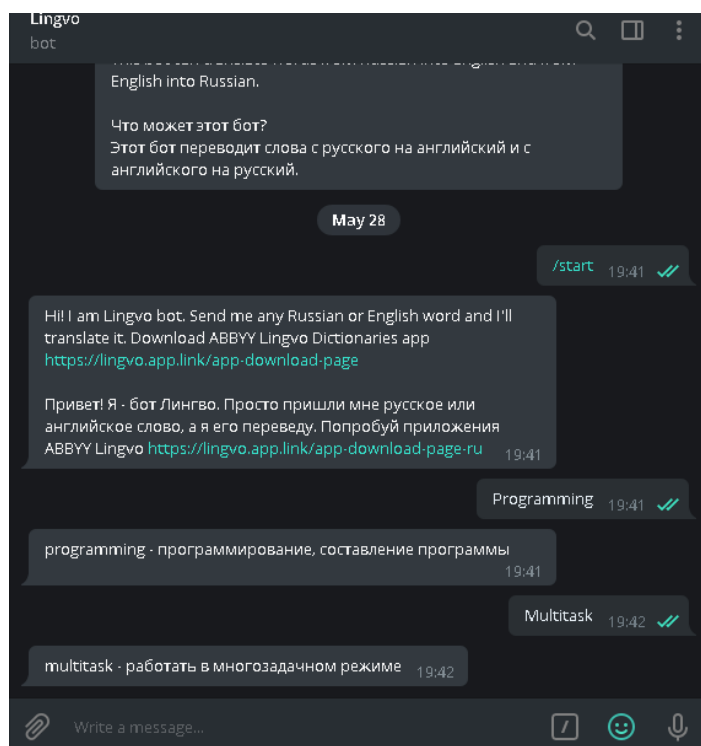


рис. 1.7 работа @LingvoBot

- @WeathersBot

Бот витягує інформацію про погоду з сайту Gismeteo.ru та передає потрібну інформацію користувачу по його запиту та геолокації(рис. 1.8).

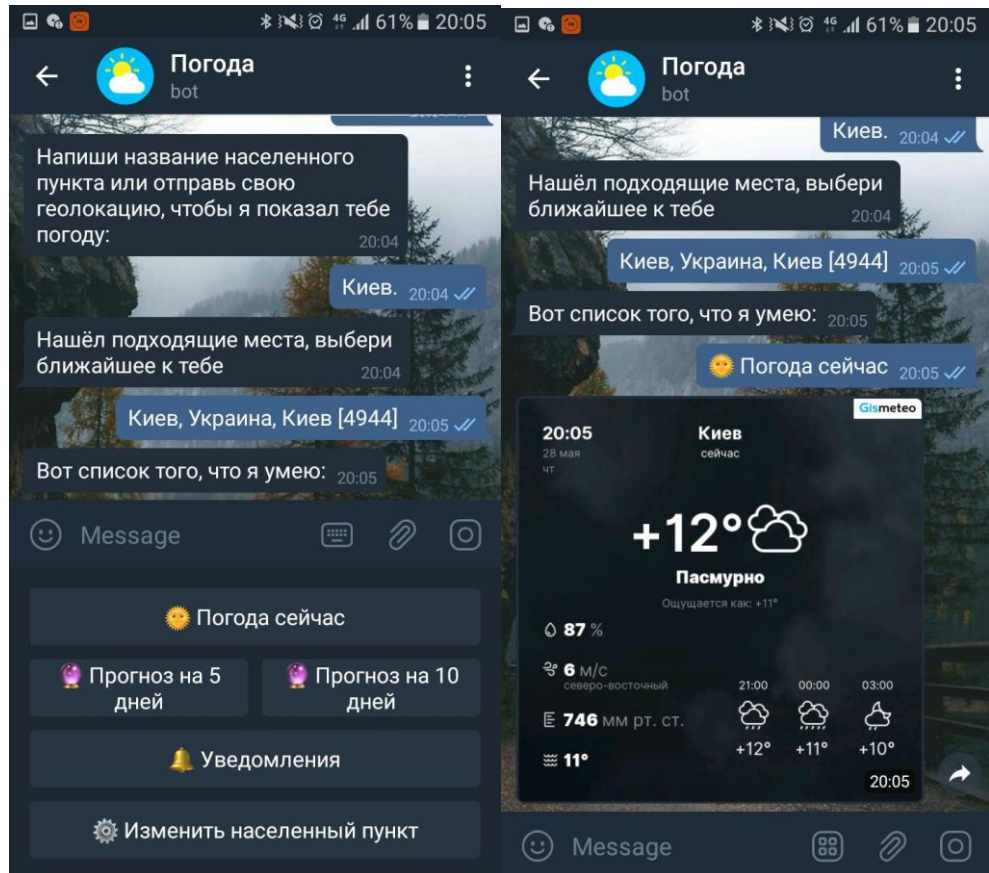


рис. 1.8 надсилання інформації про погоду @WeathersBot

ВИСНОВОК ДО РОЗДІЛУ 1

Впродовж дослідження усієї інформації по першому розділу, було отримана основна інформація про Телеграм, з'ясована сутність та функціонал телеграм ботів, її багатогранність, можливості та переваги.

Існує велика кількість ботів, їх різновидів направлень та видів, створення та направлення бота обмежується тільки уявою розробника.

Використання телеграм ботів вже давно як інтегровано в повсякдення суспільства – вони доволі поширені навіть серед великих компаній та організацій.

Створити свого телеграм-бота може кожний, якщо оволодіє деякими знаннями про Telegram Bot API.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

РОЗДІЛ 2. ВИБІР ЗАСОБІВ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ

Для створення телеграм-боту підійде будь-яка мова, яка може працювати з Telegram Bot API і має бібліотеки для праці з ним. Також існує багато конструкторів ботів для Telegram, але вони майже завжди потребують оплати за користування або містять рекламу[10].

Тому краще самому вдосконалювати свої навички в написанні бота та бути обмеженим лише своєю уявою.

2.1 Розробка на JavaScript

Спочатку JavaScript був зроблений для різнобічної праці з веб-сторінками, щоб «зробити веб-сторінки живими».

Програмний код JavaScript зазвичай називають скриптами. Вони можуть входити до HTML і запускати виконання свого скрипту при завантаженні сайту.

Скрипти JavaScript поширюються і виконуються, як простий текст. Вони не потребують спеціальної обробки або підготовки з компіляцією для запуску.

Це виділяє JavaScript від іншої схожій по назві мови – Java[11].

JavaScript надає розробнику такі функції:

- Можливість змінювати сторінки браузерів.
- Можливість додавати або видаляти теги.
- Можливість міняти стилі сторінок.
- Можливість отримувати звіт о діях користувача на сторінці.
- Працювати з cookie-файлами.

Найбільші переваги цієї мови є:

- Підтримка великої кількості різних браузерів.
- Легка взаємодія з додатками, редагувати які взагалі можна навіть використовуючи звичайні текстові редактори.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Одним з способів написання ботів в Telegram — є написання їх з використанням JavaScript, а двома основними інструментами в цьому будуть: node-telegram-bot-api I Telgraf.js.

Telegraf.js.

Telegraf.js - основна бібліотека, на якій розробнику відкриється можливість написати проект. Вона використовує певну низку підходів, які дозволяють розробнику створювати складних ботів зберігаючи простоти в коді. Нижче представлені найбільш цікаві механізми для розробника:

- Middlewares - так, це middlewares, які наявні в Express.js. Коли користувач щось відправляє боту, повідомлення по-перше проходить через всі наявні middlewares, а вже потім надходить в контролер.

Middlewares можуть змінювати контекст, а також зупиняти виконання запиту, наприклад, якщо користувач спробує потрапити в середу для адміністраторів. Telegraf.js надає можливість використання кількох вкрай корисних middlewares з запропонованих, але також можна використовувати особисті.

- Sessions - сесії, які мають можливість зберігати в собі інформацію, не зв'язану з контролером. Можна навести аналогію з глобальними змінними, які можуть використовуватися у будь-якій частині бота. Використовується, наприклад, для локалізації. Є функція зберігати сесії в різних станах - БД, redis, локальні файли і т.д.

- Webhooks - бот має працювати лише в двох режимах - long polling або Webhooks . Працюють вони однаково швидко, все ж в production середовищі рекомендується використовувати інший варіант. Тому боту не потрібно буде відправляти запити на сервер Telegram, замість цього він сам буде приймати всі оновлення.

Обов'язковою умовою для Webhook'ов є підтримка сервером SSL / TLS. Також користувач має переконатися, що порт, на якому приступний Webhook, відкритий і доступний ззовні для запитів.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

- Markup – цей клас відповідає за створення кнопок у чаті з ботом використовуючи вказивши боту відповідати в markup / markdown розмітці. Важливо для розробника робите HTML-гру. Проте заголовком його використовували для bold / italic тексту.
- Stage - модуль, який спрощує розробку боту, зменшує кількість часу та помилок в коді.

Бот вміє відгукатися на повідомлення від користувача і, в залежності від типу повідомлення користувача, перенаправляти його в певний контролер, який викличе потрібну функцію і надасть відповідь користувачеві.

2.2 Веб-система Django

Django - це веб-система Python високого рівня, яка надає швидкий шлях, пропонуючи користувачеві вже готові рішення розвитку його програмного коду, який вже був протестований та обраний певною низкою користувачей[12].

З самого початку Django був зроблений для роботи з Apache і з використанням PostgreSQL, як система управління даних. З часом Django розвинувся та зміг працювати з іншими продуктами управління БД:

- MySQL
- SQLite
- Microsoft SQL Server
- DB2
- Firebird
- SQL Anywhere
- Oracle

Для праці з ними Django має власний ORM.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

ORM - (англ. Object-Relational Mapping, Об'єктно-реляційне відображення, або перетворення) – це технологія програмування, яка зв'язує БД з концепціями ООП мов програмування, створюючи віртуальну об'єктну БД. За рахунок цього і відбувається створення схеми БД в програмному коді.

Django пропонує багато засобів для прискорення розробки веб-сайтів.

Найпоширене використання Django є побудова адмін.-панелі. Завдяки цьому замість створення панелі керування коду можна використовувати вбудований додаток, який можливо внести в будь-який сайт, побудований з Django, також є можливість моніторити стан одразу певної низки ресурсів лише з однієї панелі. Django надає можливість змінювати будь-який контент сайту, записуючи всі дії, і надає керування над групами користувачей.

Функціонал Django:

- Підключення до БД і підтримка транзакцій
- Має шаблон панелі адміністрування
- Кеш
- Має шаблони функцій
- Підключення до різних модулів аутенфікації
- Спрощення роботи з формами за рахунок вбудованих бібліотек
- Автоматична документація по тегам шаблонів.

Переваги:

1. Багатогранність.

Django можна використовувати у більшості поставлених задач — його можна застосувати майже для будь-якого сайту, завдяки набору шаблонів та їх композиції й підбору.

2. Масштабованість.

Django гарно працює при високому трафіку, завдяки чому швидко й розповсюдився серед розробників.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

3. Безпечність.

Django пропонує вже використанні та протестовані рішення проблеми, які зарекомендували себе за певний час.

4. Швидкість

Розробник, який працює з Django витрачає менше часу на пошук рішення, на рутинну працю, що значно зберігає час та прискорює роботу не знижуючи якість продукту.

Недоліки: Django занадто унітарний. Django має велику кількість шаблонів, які можна компанувати, для рішення поставленої задачі, але вони не надають певної гнучкості.

2.3 Система управління БД MongoDB та MySQL

MongoDB[13].

MongoDB – документоорієнтована СУБД з open-source кодом, яка не потребує у розробника опису схеми таблиць. Класифікована в посібниках як NoSQL(рис. 2.1).

Особливості та переваги MongoDB[14]:

- Відсутність схеми.

Це одна з найбільших переваг цієї системи. Якщо користувач має проект з великою кількістю роботи – це один з найкращих рішень. До виходу на ринок великий проект зазнає багату кількість змін, але завдяки MongoDB, вносити зміни в проект буде набагато легше. В MongoDB нема необхідності створення таблиць, змінення схем, творення міграцій, дотримуватися типів даних.

- Просте горизонтальне масштабування.

Горизонтальне масштабування потрібно коли розробнику необхідно помістити базу даних інформації більше, ніж ємкість диску на вашому сервері. Все, що пов'язано з горизонтальним масштабуванням є особливістю будь-якої NoSQL бази даних.

					ІАПЦ.467200.003 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

- Великий обсяг функцій агрегації.

Map-reduce, і угрупування непростих умов, і переробка документів впродовж усього кодування, і отримання розробником випадкових документів та сортування. Це розробник може вивчити з SQL БД, плюс можливість надавати запит в форматі pipeline'ов і з більш простим синтаксисом.

Один з прикладів:

```
sql

SELECT cust_id,
       SUM(price) as total
FROM orders
WHERE status = 'A'
GROUP BY cust_id
HAVING total > 250

mongo

db.orders.aggregate( [
  { $match: { status: 'A' } },
  {
    $group: {
      _id: "$cust_id",
      total: { $sum: "$price" }
    }
  },
  { $match: { total: { $gt: 250 } } }
] )
```

рис. 2.1 приклад коду sql та mongo

- Дозволяє зберігати дані, як потрібно розробнику.

У MongoDB загалом можливо зберігати дані, як буде зручно розробнику. У SQL базах даних завжди підтримувати організацію усіх даних, щоб таблиці були нормалізовані, а запити створювати так, щоб без помилок діставати дані з БД.

- Простий формат індексів.

Індекси в MongoDB мають просту та зрозумілу назву і їх використання практично будь-яких сюрпризів. Наприклад, в постгресе якщо у розробника є b-tree індекс на перше поле і gist індекс на друге – при надходженні запиту, який використовує обидва цих індексу, виконається тільки один з них. У Монго таких сюрпризів менше[15].

MySql[16].

MySQL це система керування БД з відкритим вихідним програмним кодом (СУРБД) з моделлю праці клієнт-сервер. СУРБД - це програмне забезпечення або служба, яка використовується розробниками для створення та керування БД на основі реляційної моделі, яка використовує відкритий код.

Відкритий вихідний код означає, що розробник може вільно використовувати його в своїх проектах та змінювати. Кожен може встановити програмне забезпечення. Клієнт також може вивчити і налаштувати вихідний код, щоб він краще відповідав його потребам. Комерційна ліцензована версія доступна, якщо розробнику потрібно більш різнобічне володіння і розширена підтримка.

MySQL є одним з великої кількості різновидів забезпечення СУРБД. Вважається, що СУРБД і MySQL мають таку ж популярність як і MySQL. Певна низка великих веб-додатків, таких як Facebook, Twitter, YouTube, Google і Yahoo! - оперують MySQL для зберігання даних. Але спочатку він розроблявся для обмеженого використання, зараз він сумісний з багатьма платформами для обчислення, такими як Linux, macOS, Microsoft Windows і Ubuntu.

SQL.

MySQL і SQL різні, незважаючи на схожу назву. MySQL є однією з поширених торгових марок програмного забезпечення СУРБД, яка працює по типу модель клієнт-сервер. Клієнт і сервер пов'язані в середовищі СУРБД. Вони оперують на специфічний для домену мові - мова структурованих запитів (SQL). Є також схожі імена з SQL, в яких є SQL, такими як PostgreSQL і сервер Microsoft SQL, вони, зазвичай, є брендами, які також використовують синтаксис SQL. Програмне забезпечення СУРБД часто розробляється на інших мовах програмування, але завжди використовує SQL, як основну мову для взаємодії з БД. Але сам MySQL написаний за допомогою C і C ++.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Інженер-комп'ютерник Тед Кодд створив SQL на початку 1970-х років використовуючи за основу реляційну модель IBM. Він поширюватися більше в 1974 році і доволі швидко та без зусиль замінив аналогічні, тоді застарілі мови, ISAM і VISAM. Крім історії, SQL може повідомляти серверу, що робити з інформацією. Клієнт додає його в систему, щоб отримати доступ до панелі управління. При цьому розвитку подій оператори SQL можуть вказати серверу виконати певну низку операцій:

- Запит даних:

Запит конкретної інформації з існуючої БД.

- Обробка даних:

Функції додавання, видалення, зміна, сортування та інші операції для створення інших даних, значень або візуальних елементів.

- Ідентифікація даних:

Визначення певних типів даних, наприклад, зміна числових даних в цілі числа. Це також включає визначення схеми або взаємозв'язку будь-якої таблиці в БД.

Контроль доступу до інформації: створення методів безпеки для захисту даних, у тому числі дозвіл перегляду або використовувати будь-яку інформацію, що існує в даній БД.

Кожен бажаючий клієнт може зробити запит з графічного інтерфейсу користувача (GUI) на своїх моніторах, і сервер відрпаве шукаємий результат, якщо обидва кінці розуміють вхідні дані. Не вдаючись в технічні аспекти, основні процеси, що проходять в середовищі MySQL, однакові:

MySQL створює БД для зберігання і управління інформацією, що визначають відносини кожної таблиці у БД.

Клієнтам надана можливість робити запити, використовуючи певні команди SQL на MySQL.

Додаток сервера відповість шукаємою інформацією і клієнт побачить його на своїй стороні.

					ІАПЦ.467200.003 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

Клієнти майже завжди вказують, який MySQL GUI використовувати. Чим легше і зручніше графічний інтерфейс для користувача, тим швидше і простіше будуть оброблятися операції з управління даними. Деякими з розповсюджених графічних інтерфейсів MySQL є WorkBench, SequelPro, DBVisualizer і Navicat DB Admin Tool. Частина з них безкоштовні, частина комерційні, а деякі працюють тільки для macOS, а деякі сумісні з основними операційними системами. Клієнти мають вибирати графічний інтерфейс в залежності від своїх запитів. Для управління БД на сайті WordPress, найбільш зручним рішенням є phpMyAdmin.

MySQL не єдина СУРБД на ринку, але вона є однією з більш розповсюджених і програє тільки Oracle Database, коли оцінюється з використанням таких важливих якостей, як кількість згадок в результатах пошуку даних, професійних профілів в LinkedIn і частоти технічних дискусій на інтернет-форумах. Той факт, що багато великих компаній покладаються на нього, ще більше зміцнює позицію. Ось причини:

- Гнучкість і простота у використанні

Розробник може змінювати вихідний код, щоб він відповідав власним очікуванням та потребам розробника, і йому не потрібно нічим поступатися за цей рівень свободи, включаючи різні варіанти оновлення до розширеної комерційної версії. Процес установки дуже простий і не повинен потребувати більше 30 хвилин часу.

- Висока продуктивність

Різноманітність кластерних серверів підтримує MySQL. Незалежно від того, чи зберігає клієнт великі обсяги інформації електронної комерції або обробляє важку бізнес-аналітику, MySQL може допомогти клієнту з налагодженням потрібної швидкості.

- Промисловий стандарт

Різні великі й малі компанії використовують MySQL протягом багатьох років, для досвідчених розробників є великі ресурси. Користувачі MySQL можуть отримати швидку розробку програмного забезпечення, а експерти-

					ІАПЦ.467200.003 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

фрілансери, які хочуть працювати в меншій мірі, будуть мати потребу в них.

- **Безпека**

Безпека клієнтських даних повинна бути головним завданням при виборі найкращого програмного забезпечення СУРБД. За допомогою системи доступу і керування обліковими записами MySQL встановлює високий рівень безпеки. Є можливість перевірки на основі хоста і шифрування пароля.

Якщо говорити про MySQL - це перевірена технологія. Авжеж MySQL використовується великими різними компаніями більше 15 років. Так як він використовує стандарт SQL, є можливість досить простих міграцій на інші SQL-бази даних, якщо потрібно. Є можливість транзакцій. Підтримуються складні запити, включаючи аналітику.

Дивлячись на MongoDB, тут перевага в тому, що він має гнучкий JSON-формат документів. Для деяких завдань і деяким розробникам це зручніше, ніж додавання колонок в SQL-базах даних. Прості запити менше створюють проблеми. Якщо подивитися на проблеми продуктивності, в основному вони виникають, коли люди створюють складні запити з JOIN до купи таблиць і GROUP BY. Якщо така функціональність в системі відсутня, то створити складний запит виходить складніше.

2.4 Шляхи оновлення бота: Webhook та getUpdates

На даний момент в Телеграм є два способи оновлення телеграм-боту[17]:

- Long Polling.
- setWebhook[18].

Коли розробник пише telegram бота і запускає його без вебхуків, то наш бот (який крутиться на нашому сервері / комп'ютері) постійно звертається до серверів телеграма(рис. 2.2) і питає чи є для нього інформація(Long Polling). Іноді сервера телеграма не відповідають і наш бот перестає працювати. Звичайно при такому розкладі буде періодично простій бота в 2-15 сек і наш бот буде постійно звертатися до серверів телеграма, створюючи на них

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

навантаження.

Для збільшення швидкості роботи бота, відмовостійкості і прибирання створеного навантаження використовують вебхук. При роботі бота на вебхуках сервер телеграма сам звертається до нашого боту коли є нові дані. Власне і на наш сервер навантаження буде менше, адже бот частіше буде простоювати, що розглянемо як безсумнівний плюс(рис. 2.3).

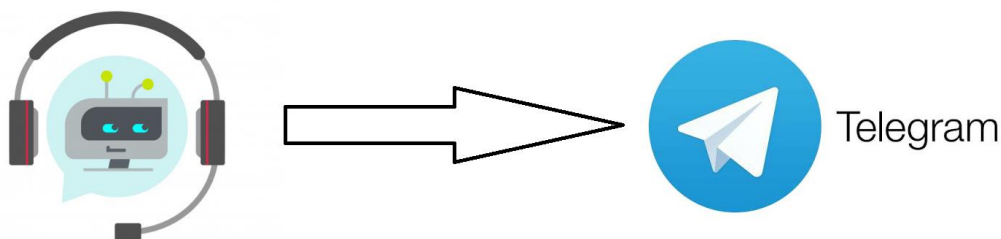


рис. 2.2 Long Polling

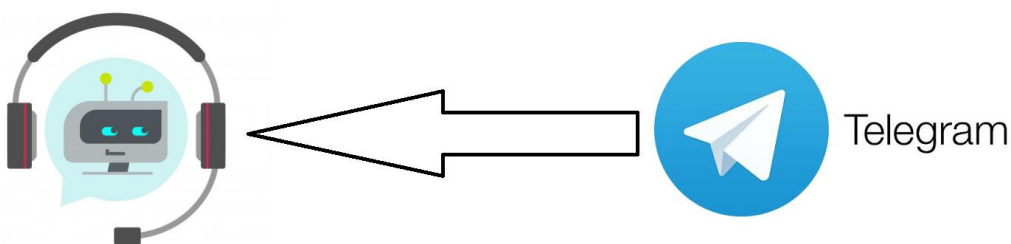


рис. 2.3 setWebhook

Обидва способи одночасно (вебхук і поллінг) використовуватися не можуть, - якщо встановлений вебхук, то відповідати на запити сервери телеграм НЕ будуть.

Клієнту при використанні поллінгу замість вебхуків не потрібна зворотня адреса (хостинг і доменне ім'я), тобто можна хоч на домашньому комп'ютері Telegram-бота підняти. Є і недоліки, але додаткові складності, - апдейти тепер будуть прилітати не по одному, а масивами, відповідно, спочатку потрібно буде розібрати отримані дані на окремі апдейти, а потім вже обробляти їх аналогічно тому, як це робиться з вебхуками.

API telegram для роботи через поллінг.

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Отже, для отримання масиву апдейтів потрібно скористатися методом `getUpdates` і відправити на сервера телеграм запит наступного виду: [https://api.telegram.org/botYOUR_BOT_TOKEN/getUpdates\[?options\]](https://api.telegram.org/botYOUR_BOT_TOKEN/getUpdates[?options]), де `YOUR_BOT_TOKEN` - токен вашого бота, `options` - список додаткових опцій.

Опції можуть бути наступними:

- `offset` - ідентифікатор першого запитуваного апдейта. З цією функцією будуть отримані всі апдейти, починаючи з апдейта, який з зазначеним в поле `offset` ідентифікатором. Тобто, для того, щоб отримувати всі апдейти – розробник повинен кожен раз в запиті вказувати ідентифікатор на одиницю більше, ніж ідентифікатор (`update_id`) останнього отриманого апдейта. Всі апдейти з ідентифікаторами менше отриманого значення `offset` «забуваються»;
- `limit` - максимальна кількість апдейтів, яке буде отримано за один раз. Може приймати значення від 1 до 100 (за замовчуванням 100).
- `timeout` - час очікування для «довгого» запиту. За замовчуванням = 0 (тобто за замовчуванням обирається звичайне опитування);
- `allowed_updates` - тип повідомлень, які розробники хотіли б отримувати (наприклад, типу `message`). Ця опція додана недавно, тому інколи не вдається домогтися її заявленої роботи (вибрані типи повідомлень приходять не від усіх користувачів).

2.5 Розробка на Python

Для розробки телеграм-ботів на пайтоні використовують 2 бібліотеки та засоби для зв'язання пайтону[19] з телеграм-ботами(`AIOGram`[20], `Telepot`, `Telegram Bot Service`, `telebot`, `twx.botapi`):

- `pyTelegramBotAPI`
- `Python-telegram-bot`

`PyTelegramBotAPI`[21].

Загальні функції цього API це[22]:

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

- Обробка повідомлень.

Обробник повідомлень - це функція, прикрашена `message_handler` декоратором екземпляра `TeleBot`. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає значення `True` для певного повідомлення, щоб обробник повідомлень мав право на обробку цього повідомлення. Обробник повідомлень оголошується наступним чином (надається `bot` екземпляр `TeleBot`):

```
@ bot.message_handler ( фільтри )
```

```
def function_name( message ):
```

```
    bot.reply_to ( повідомлення , «Це обробник повідомлень» )
```

`function_name` не пов'язаний з жодними обмеженнями. Будь-яке ім'я функції дозволено в обробниках повідомлень.

Функція повинна приймати щонайменше один аргумент, який буде повідомленням, яке функція повинна обробляти. `filters-` це список аргументів ключових слів. Фільтр оголошуються в такий спосіб: `name=argument`. Один обробник може мати кілька фільтрів. `TeleBot` підтримує такі фільтри:

- Обробка запитів зворотного дзвінка.

В оновленнях `bot2.0`. Ви можете потрапити `callback_query` в об'єкт оновлення. За допомогою телеботів `callback_query_handler` для обробки запитів зворотного виклику.

- Обробка проміжного програмного забезпечення.

Обробник програмного забезпечення - це функція, яка дозволяє змінювати запити або контекст бота під час передачі через Telegram до бота. Ви можете уявити проміжне програмне забезпечення як ланцюжок логічного з'єднання, що обробляється перед виконанням будь-яких інших обробників.

- `TeleBot`

					ІАПЦ.467200.003 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- Розмітка відповідей
- Inline Mod

Також ця бібліотека має розширене використання API:

- Асинхронна доставка повідомлень

Існує реалізація TeleBot, яка виконує всі `send_xyz` та `get_me` функції асинхронно. Це може значно пришвидшити роботи бота, але він має небажані побічні ефекти, якщо використовувати їх обережно. Щоб увімкнути цю поведінку, треба створити примірник AsyncTeleBot замість TeleBot.

```
tb = телебот . AsyncTeleBot ( "TOKEN" )
```

Тепер кожна функція, яка викликає API Telegram, виконується в окремій потоці. Функції модифікуються для повернення екземпляра AsyncTask (визначеного `util.py`).

- Надсилання великих текстових повідомлень

Іноді потрібно надсилати повідомлення, що перевищують 5000 символів. API Telegram не може обробити стільки символів в одному запиті, тому нам потрібно розділити повідомлення на кілька.

- Контроль кількості потоків, якими користується TeleBot

Конструктор TeleBot приймає такі необов'язкові аргументи:

`stringed: True / False` (за замовчуванням `True`). Прапор, який вказує, чи повинен TeleBot виконувати обробники повідомлень на опитувальній нитці.

- Механізм слухача

Як альтернатива обробникам повідомлень, можна також зареєструвати функцію як слухача TeleBot.

Обробники не зникнуть. Повідомлення клієнта буде оброблятися як обробниками, так і слухачами. Також неможливо передбачити, що спочатку спрацює через різьблення. Якщо ви використовуєте `threaded = False`, користувацькі слухачі працюватимуть раніше, після цього вони будуть викликані обробникам.

- Використання вебхуків.

Під час користування вебхуком, телеграм надсилає одне оновлення за виклик,

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

для його обробки слід викликати `process_new_messages ([update.message])`, коли ви отримуєте його.

У каталозі прикладів / `webhook_examples` є кілька прикладів використання вебхуків .

- Ведення журналів

Розробник може використовувати реєстратор модулів Telebot для реєстрації інформації про налагодження про Telebot. Використовуйте `telebot.logger` для отримання реєстратора модуля TeleBot. Можна додати користувальницькі обробники журналів до реєстратора. Докладнішу інформацію див. На сторінці модуля журналу Python.

- Проксі

Розробник може використовувати проксі для запиту. `apihelper.proxu` об'єкт буде використовувати `requests` аргумент проксі-сервера виклику.

					ІАПЦ.467200.003 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК ДО РОЗДІЛУ 2

Обробивши велику кількість доступної інформації про створення телеграм-ботів, їх сильні та слабкі прояви, використовуючи різні бібліотеки, БД, мови програмування та інше, було обрано шлях написання боту на мові програмування Python для даного дипломного проекту. Ця мова є більш простою та гнучкою.

					ІАПЦ.467200.003 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ НА МОВІ ПРОГРАМУВАННЯ PYTHON

В цьому розділі для реалізації дипломного проекту було використано мову програмування Python та середовище для програмування IDE PyCharm.

3.1 Мова програмування Python

Мова програмування Python являє собою інтерпретовану ООП(об'єкто-орієнтоване програмування) мову програмування високого рівня та несе в собі строгу динамічну типізацію даних. Має в собі високорівневі структури даних, які у купі з динамічною семантикою та динамічним зв'язуванням надають можливість швидкої розробки програм. Мова програмування Python також підтримує модулі та пакети модулів, що надає можливість розробнику використати програмний код повторно. В мові Python підтримується деяка низка парадигм програмування:

- ООП.
- Процедурна.
- Аспектно-орієнтована.
- Функціональна.

Парадигма програмування – являє собою систему ідей та понять, які задають стиль написання програмного коду, а також спосіб вирішення конкретних задач, опираючись на задану парадигму.

ООП(об'єктно-орієнтоване програмування) – сутність цієї парадигми – розглядати увесь програмний код написаний розробником – як множину об'єктів, що взаємодіють між собою. Ця парадигма має за основу чотири концепції написання:

- Інкапсуляція.

Об'єкт має не тільки дані, але також має власні правила їх обробки, які представлені як виконувані фрагменти(або ж методи). Також доступ до

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

об'єкту напряму заборонено, а ззовні з ним можна взаємодіяти тільки через заданий інтерфейс – це дозволяє знизити зв'язність у програмному коді. Завдяки цьому розробник може контролювати звернення до полів класів та їх правильну ініціалізацію, усуваючи можливі помилки, які пов'язані з неправильним викликом методу(рис. 3.1).

```

1 class SomeClass:
2     def _private(self):
3         print("Це внутрішній метод об'єкту")
4
5     obj = SomeClass()
6     obj._private() # Це внутрішній метод об'єкту

```

Run: main (1) ×

C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py
 Це внутрішній метод об'єкту
 Process finished with exit code 0

```

1 class SomeClass():
2     def __init__(self):
3         self.__param = 42 # захищений атрибут
4
5     obj = SomeClass()
6     obj.__param # AttributeError: 'SomeClass' object has no attribute '__param'
7     obj.SomeClass__param # 42

```

Run: main (1) ×

C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py
 Traceback (most recent call last):
 File "C:/Users/1/PycharmProjects/PyWorking/main.py", line 6, in <module>
 obj.__param # AttributeError: 'SomeClass' object has no attribute '__param'
 AttributeError: 'SomeClass' object has no attribute '__param'
 Process finished with exit code 1

рис 3.1 приклад інкапсуляції в Python

- Успадкування.

Це спосіб створення нових класів, використовуючи вже існуючі класи в програмному коді. При цьому дочірній клас, який наслідує материнський, успадковує та зберігає усі властивості та функціональність(рис 3.2). Новий клас отримує поведінку класу, який наслідує та доповнює його своїми власними властивостями та функціоналом.


```

1 class Simple():
2     className = 'Simple'
3
4 class Dog(Simple):
5     species = 'Exapmle'
6
7 dog = Dog()
8 print(dog.className) # Simple
9

```

Run: main (1) ×

C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py

Simple

Process finished with exit code 0

рис. 3.2 приклад наслідування у мові програмування Python

Застосування механізму успадкування надає можливість розробнику покращити використання коду у наступні рази, при цьому зберігаючи цінний час, шляхом використання можливостей базових класів створених завчасно.

- Поліморфізм.

Поліморфізм являє собою концепцію в програмуванні коду на теорії типів, на основі якої для обробки різних спеціалізованих типів об'єктів використовується спільний інтерфейс(рис. 3.3).

```

1 class English:
2     def greeting(self):
3         print("Hello")
4
5
6 class French:
7     def greeting(self):
8         print("Bonjour")
9
10
11 def intro(language):
12     language.greeting()
13
14 flora = English()
15 aalase = French()
16
17 intro(flora)
18 intro(aalase)
19

```

Run: main (1) ×

C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py

Hello

Bonjour

Process finished with exit code 0

рис. 3.3 приклад поліморфізму в Python

- Абстракція.

Для цієї концепції Python має свою стандартну бібліотеку – Abstract Base Classes. У абстрактному класі з'ясовується – які методи та властивості розробнику треба буде переобрати в дочірніх класах.

```

1 class Salary(object):
2     def __init__(self, pay):
3         self.pay = pay
4
5     def getTotal(self):
6         return (self.pay * 12)
7
8 class Employee(object):
9     def __init__(self, pay, bonus):
10        self.pay = pay
11        self.bonus = bonus
12
13    def annualSalary(self):
14        return "Total: " + str(self.pay.getTotal() + self.bonus)
15
16 salary = Salary(100)
17 employee = Employee(salary, 10)
18 print(employee.annualSalary())

```

Run: main (1) ×

C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py

Total: 1210

Process finished with exit code 0

рис. 3.4 приклад абстракції(асоціації) в мові програмування Python

Процедурна парадигма програмування – створена на основі виклику процедури, також відомі, як підпрограми або методи, функції. Кожна з процедур містить деяку послідовність кроків для виконання. Особливість полягає в тому, що в будь-якій частині програмного коду – є можливість виклику процедури, навіть включно з самої процедури.

Переваги:

- Має можливість повторного використання строк коду з різних місць програми, не дублюючи сам код.
- Набагато легше розробнику відстежити хід виконання програмного коду.
- Має можливість підтримку модульності та структурності.

Аспектно-орієнтована парадигма програмування(АОП) дозволяє розробнику виділити перехресну функціональність у програмному коді(cross-cutting concern).

Цю парадигму частіше використовують для:

- Ведення журналів подій.
- Керування винятковими ситуаціями у коді.
- Перевірки прав доступу до програмного коду.
- Відстеження коду, який відповідає за певну функціональність, який часто розкиданий по різних класах.

Також ця парадигма може бути використана розробником для проектування будь-яких систем на основі наскрізної функціональності програмного коду. Виходячи з цього, ця парадигма має певні недоліки:

- Розкиданий код, тому що наскрізна функціональність охоплює велику кількість модулів системи й виклики цієї функціональності будуть розкидані по всьому програмному коду.
- Важко відслідковувати за призначеним розробником модулем, так як він має в один час функціональність для реалізації різних вимог.
- Велика складність або повна неможливість повторне використання розробником модуля у програмному коді з іншою наскрізною функціональністю.
- Велика ймовірність помилок. Через те що в коді є реалізації функціональності різного роду і все в одному модулі – це може призвести до того, що жодне з багатьох завдань розробника не буде виконано.
- Важкість зміни модулів. Якщо у розробника виникає потреба в зміні наскрізної схеми функціональності програмного коду то така зміна зачіпає багато модулів системи, що зазвичай приводить до створення проблеми сумісності між модулями.

Переваги:

					ІАПЦ.467200.003 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

- Увесь програмний код – набір модулів. Аспектно-орієнтований підхід дозволяє розробнику розглядати програмну систему як набір певних модулів, кожен з яких несе за собою функціонування системи.
- Різнобічність. При використанні об'єктно-орієнтованого підходу не завжди у розробника є можливість виконати поставлене завдання лише в одному окремому модулі – у результаті чого функціональність системи буде знаходитися у різних модулях(наприклад ведення розробником журналу подій у коді).
- Програмного коду менше ніж в ООП в цілому на 10-40%, завдяки чому система витрачає менше ресурсів і тим самим – часу.

Функціональна парадигма програмування – парадигма, яка представляє розробнику програму як вичислення математичних функцій при цьому уникаючи станів та змінних даних(рис. 3.5). Функціональне програмування фокусується на застосуванні тільки функцій, навпаки від концепції імперативного програмування, яке зосереджується на змінних в стані та виконанні розробником послідовних команд.

```

main.py
1  def addClosure(val1):
2      def closure(val2):
3          return val1 + val2
4      return closure
5
6  class AddFuncion(object):
7      def __init__(self, val1):
8          self.val1 = val1
9      def call(self, val2):
10         return self.val1 + val2
11
12  cl = addClosure(2)
13  fn = AddFuncion(2)
14
15  print(cl(1), fn(1)) # Програма видасть значення "3 3"

Run: main (1)
C:\Users\1\PycharmProjects\PyWorking\venv\Scripts\python.exe C:/Users/1/PycharmProjects/PyWorking/main.py
3 3
Process finished with exit code 0

```

рис. 3.5 приклад вирішення програми шляхом функціонального програмування

3.2 Середовище розробки IDE PyCharm

Для більш якісного, швидкого та структурного коду написання дипломного проекту була використана інтегрована середа розробки для мови програмування Python, а саме – IDE PyCharm, розроблений компанією JetBrains на основі IntelliJ IDEA. Підтримується такими операційними системами як:

- Windows
- MacOS
- Linux

PyCharm має багато корисних для розробника якостей, які покращують, прискорюють та вдосконалюють написання коду:

- Постійний аналіз коду, підсвічування та звертання уваги розробника на неправильний синтаксис та помилки в програмному коді мови програмування. PyCharm надає розробнику вбудований відладник, вбудований термінал, низку інструментів для обробки даних та праці з різними БД, перелік інструментів для запусків тестів, та інструменти для розробника, які дозволяють контролювати інтеграцію поширеними системами контролю версій(рис. 3.6).
- Швидка навігація. Надає можливість швидко переходити з одного проекту – на інший. Легка навігація по сирцевому коду, моніторить структуру файлів проектів.
- Можливість рефакторингу програмного коду. Швидке додавання методів, змінних, функцій, швидка їх обробка та витяг з коду.
- Має можливість використання інструментів для веб-розробки. Працює з такими фреймворками як:
 - Django
 - Flask
 - Google App Engine

- Pyramid
- Web2py

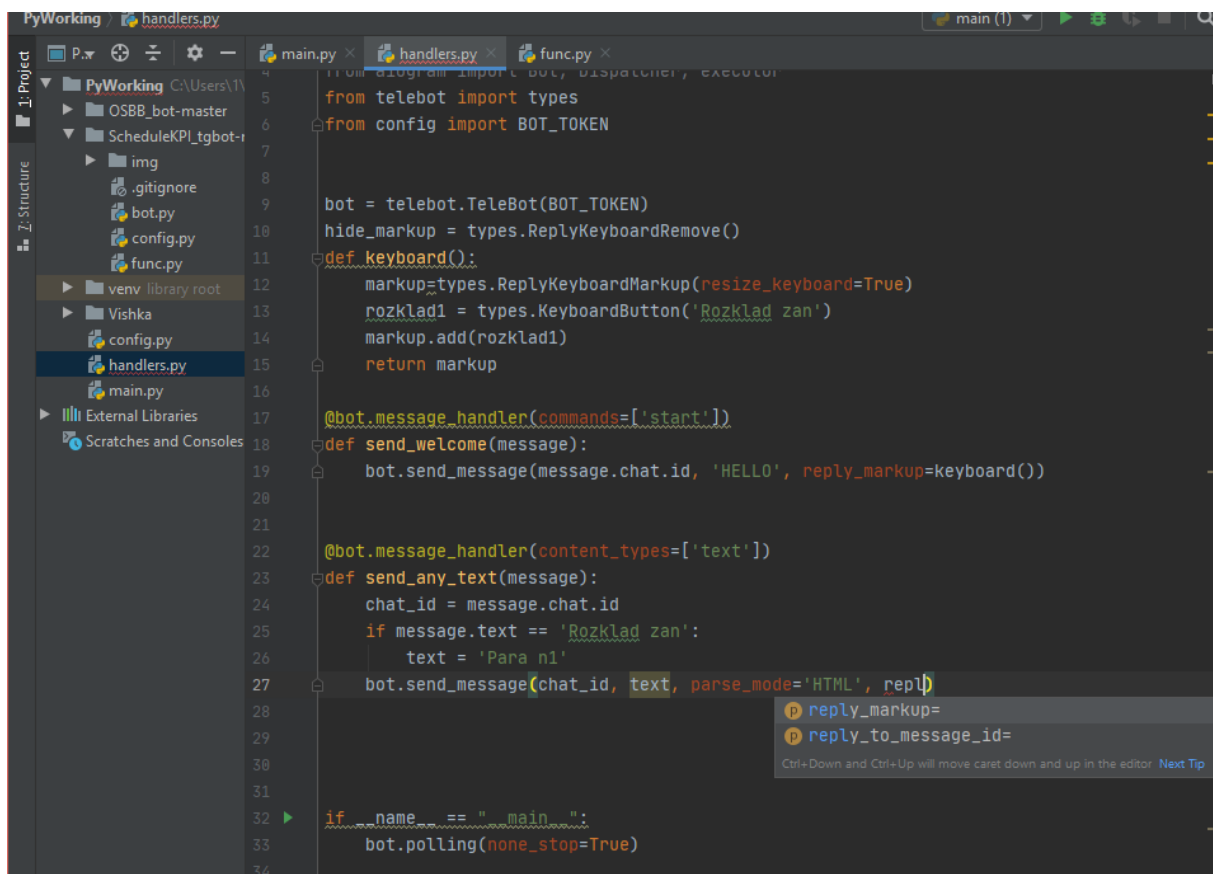


рис. 3.6 приклад праці відладника PyCharm

- Надає можливість віддаленої розробки. На PyCharm розробник може тестувати та розробляти додатки на віддалених хостах, або використовуючи різні віртуальні машини. Все це виконується завдяки віддаленим інтерпретаторам стандартного вбудованого SSH-терміналу і праці з Docker і Vagrant.
- PyCharm підтримує великі наукові обчислення. Використовуючи PyCharm розробник зможе працювати з Jupyter-ноутбуками, виконувати команди в інтерактивній консолі Python. Може також працювати з бібліотекою Anaconda, і не тільки, також використовує й інші бібліотеки для наукових обчислень, обробки та аналізу даних, використовуючи також NumPy і Matplotlib.

ВИСНОВОК ДО РОЗДІЛУ 3

Використовуючи дані з третього розділу можна зробити висновки, що завдяки таким сильним сторонам мови програмування Python та парадигмам, яких вона дотримується, а це: об'єктно-орієнтоване програмування, процедурна, аспектно-орієнтована, функціональна парадигми, буде набагато легше створити телеграм-бота з використанням меншої кількості коду.

А завдяки середі розробки IDE PyCharm буде значно легше проводити роботу над помилками, вдосконалювати код, вносити корекції та вивчати більше нових функцій та можливостей бібліотек телеграм-ботів.

					ІАПЦ.467200.003 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. ОПИС ВИКОРИСТАНИХ ФУНКЦІЙ ТА ФУНКЦІОНАЛЬНІСТЬ ТЕЛЕГРАМ-БОТУ

При створенні даного дипломного проекту була використана певна низка бібліотек Python, головними з яких є:

- Aiogram
- pyTelegramBot
- requests
- telebot
- telegram
- beatifulsoup4

4.1 Створення телеграм-боту. Опис використаних функцій

Перед тим, як почати роботу з телеграм ботом – треба його створити. Створювати бота в телеграм можна лише у іншого бота, а саме у @BotFather(рис. 4.1).

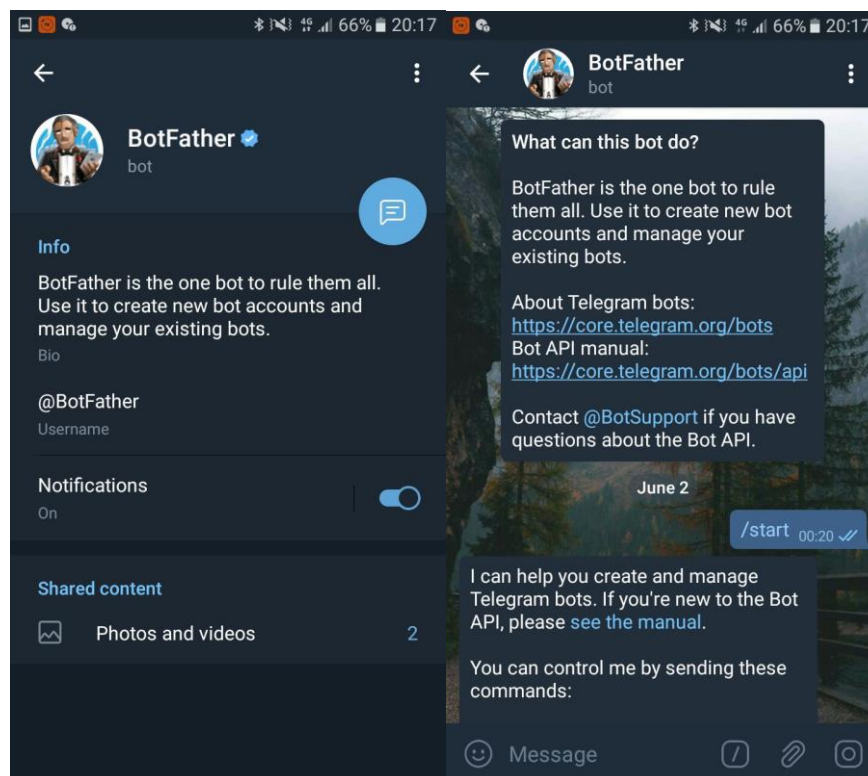


рис. 4.1 бот ля створення ботів – @BotFather

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

Вводимо команду /newbot і за допомогою зручного меню створюємо телеграм-бота, отримуючи найважливіше – його токен(рис 4.2).

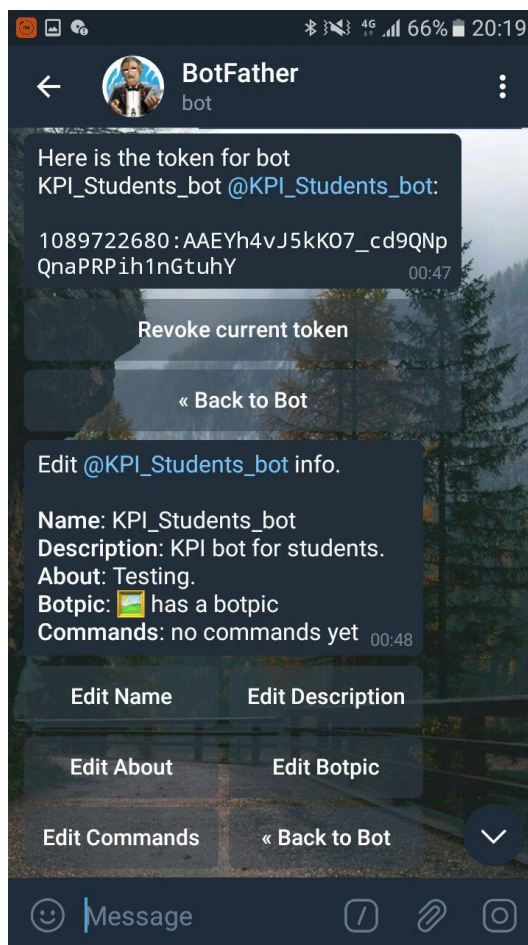


рис. 4.2 отримуємо токен від @BotFather.

Токен являє собою адресу, по якій можна програмувати телеграм-бота. Токен – це приватна інформація, так як кожний, хто має токен стороннього боту – може його змінювати. Але у разі його загублення – завжди можна спитати @BotFather токен одного з ваших ботів.

Також у налаштуваннях @BotFather розробник може змінити(рис. 4.2):

- Ім'я боту.
- Його тег.
- Його картинку на заставці(рис. 4.3).
- Описання боту та його інфо(рис. 4.3).
- Команди боту.
- А також отримати новий токен, або знайти загублений.

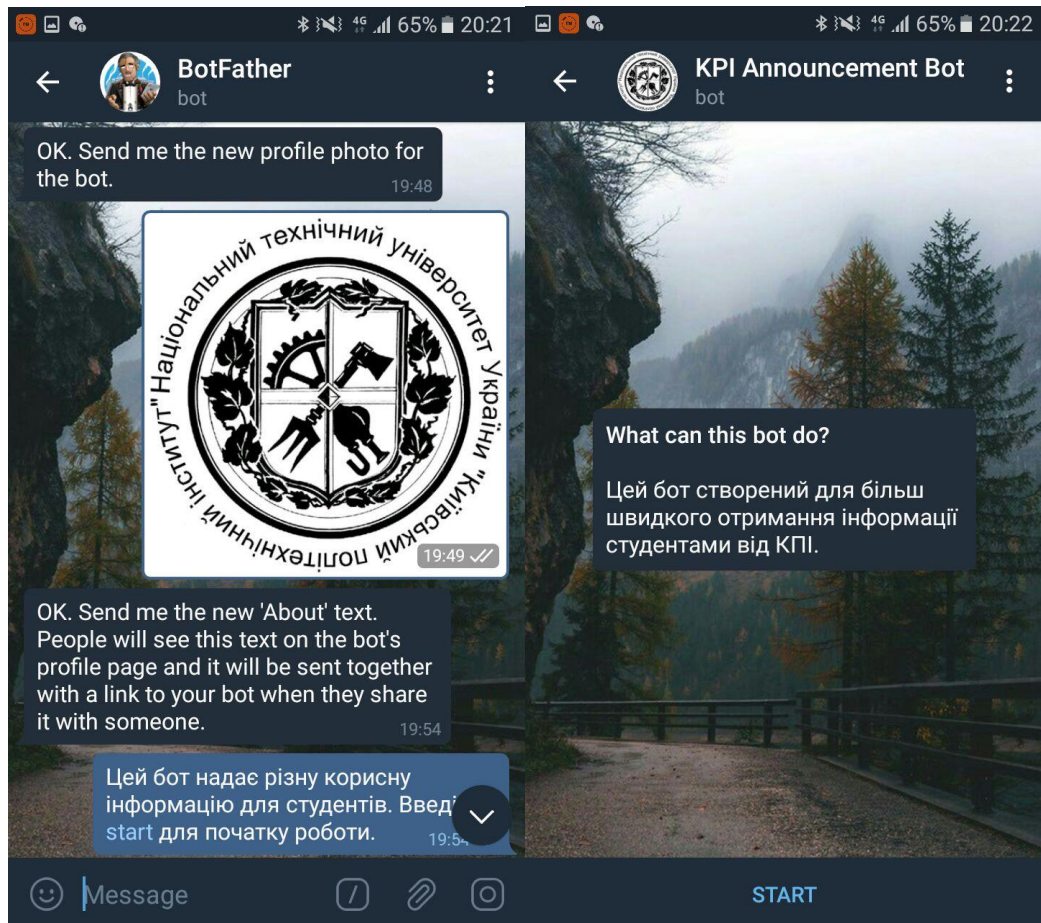


рис. 4.3 відформатований зовнішній вигляд дипломного проекту – телеграм-бота

4.2 Застосунок для миттєвого оповіщення студентів

Отримавши токен бота розробник вже може починати його програмування та налаштування на будь-які цілі. Перше, що треба налаштувати в боті – це команду /start. У цьому допоможуть хендлери. Але спочатку треба змінній надати значення бота(рис. 4.4):

```

main.py x config.py x
1 import telebot
2 from config import BOT_TOKEN
3
4 bot = telebot.TeleBot(BOT_TOKEN)

```

Рис. 4.4 присвоєння змінній значення бота за допомогою точена

В config.py будуть зберігатися сталі величини, такі як токен – вводимо в файлі:

```
token = 'Вставляємо отриманий токен від @BotFather'
```

Для початку треба розробити стандартне меню для користувача інтерфейсу. За допомогою параметру reply_markup можна виводити меню кнопок для користувача(рис. 4.5):

```
@bot.message_handler(commands=['start'])
def welcome(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=False)
    item1 = types.KeyboardButton("Розклад сесії")
    item2 = types.KeyboardButton("Розклад для викладачів")
    item3 = types.KeyboardButton("Розклад занять")
    item4 = types.KeyboardButton("Погода")
    item5 = types.KeyboardButton("Кампус")
    item6 = types.KeyboardButton("Методичні вказівки")
    markup.add(item1, item2)
    markup.add(item3, item4)
    markup.add(item5, item6)
    bot.send_message(message.chat.id,
                      "Оберіть цікаву для вас категорію:".format(
                          message.from_user, bot.get_me()),
                      parse_mode='html', reply_markup=markup)
```

рис. 4.5 створення меню кнопок для користувача в телеграм-боті

Кожна змінна itemX – це створена кнопка, їх потрібно шість. За допомогою параметрів resize_keyboard=True та one_time_keyboard=False робимо розмір кнопок трохи менший та, при натисканні на кнопки – вони не будуть зникати. Тобто меню буде головним та завжди можна буде користувачу з нього почати.

За допомогою методу .add(<Button1>, <Button2>) у телеграм-боті формується стовпчики та строки таблиці кнопок(рис. 4.6):

```
markup.add(item1, item2)
markup.add(item3, item4)
markup.add(item5, item6)
```

рис. 4.6 формування таблиці 3x3 з кнопок в телеграм-боті

Виводити цю таблицю кнопок допомагає команда `bot.send_message()`. PyCharm допомагає розібратися в існуючих параметрах для той чи іншої команди(рис. 4.7):

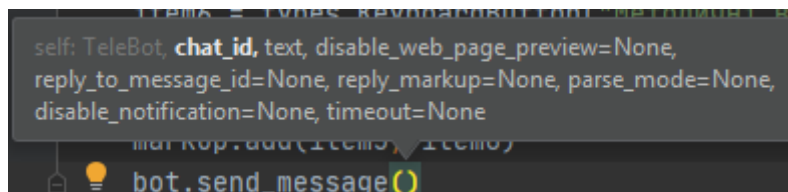


рис. 4.7 підказки від PyCharm

Використовуючи хендлер `@bot.message_handler(commands=['start'])` задаємо боту реакцію на команду `/start`. Таким же чином можна задати реакцію на будь-яку команду.

Запустивши виконання коду, запускаємо телеграм-бота. Кожен телеграм-бот завжди в початку роботи з користувачем пропонує натиснути `START` – це і буде команда `/start` та висвітить меню телеграм-бота з текстом привітання(рис.4.8):

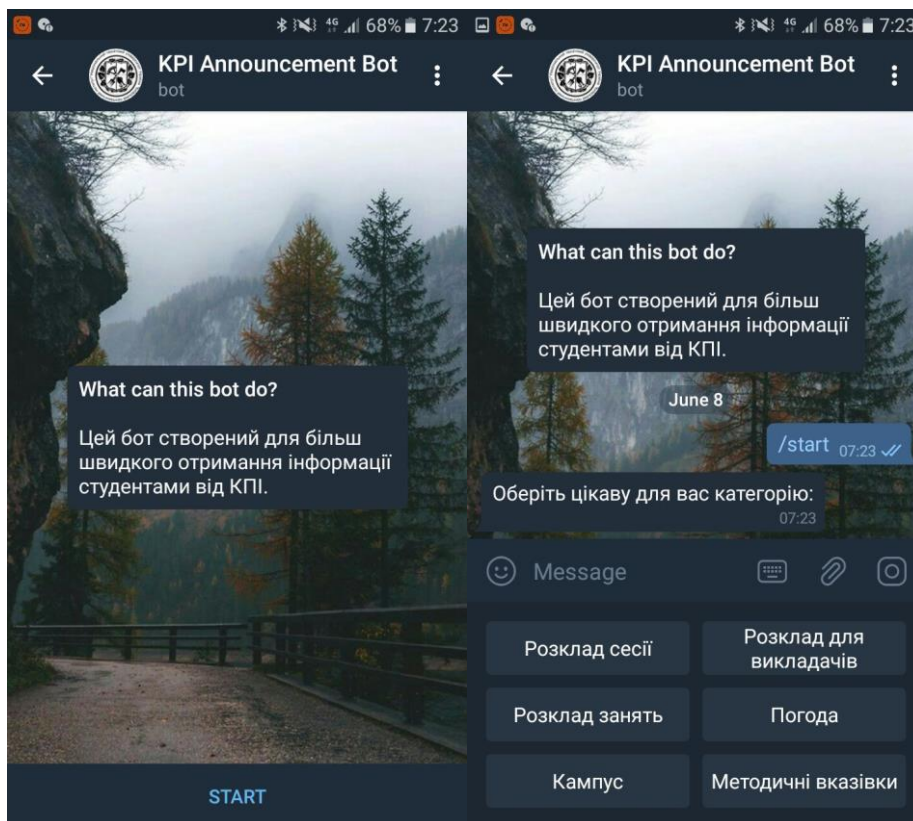


рис. 4.8 отримання меню кнопок від телеграм-боту

Майже в усіх наступних етапах дипломного проекту треба буде використати парсинг.

Парсинг – це збирання та систематизування інформації з різних сайтів для подальшої її обробки та аналізу, за допомогою програм, що роблять цей процес автоматичним. Для парсингу у Python використовується бібліотека BeautifulSoup. Ця бібліотека дозволяє парсити HTML та XML документи.

За допомогою цієї бібліотеки та бібліотеки requests, яка призначена для обробки запитів з сайтів, створюємо функцію моніторингу погоди по Києві(рис. 4.9):

```
r = requests.get('https://sinoptik.ua/погода-киев')
html = BS(r.content, 'html.parser')

for el in html.select('#content'):
    t_min = el.select('.temperature .min')[0].text
    t_max = el.select('.temperature .max')[0].text
    text = el.select('.wDescription .description')[0].text

@bot.message_handler(content_types=['text'])
def send_any_text(message):
    chat_id = message.chat.id
    if message.text == 'Погода':
        bot.send_message(message.chat.id, "Погода на сьогодні:\n" +
                          t_min + ', ' + t_max + '\n' + text)
```

рис. 4.9 створення моніторингу погоди по Києву

Моніторинг погоди було вирішено парсити з сайту <https://sinoptik.ua/> надаючи змінній r адрес сайту звідки будемо витягувати інформацію. Щоб знайти потрібну інформацію та витягнути її з сайту – треба відкрити програмний код розробника(команда Ctrl + Shift + I в браузері). Кожна частина сайту розділена на різні зони, які й містять потрібну інформацію. Треба водити курсором по класам в програмному коді браузера, щоб знайти потрібний відділ з інформацією(рис. 4.10):

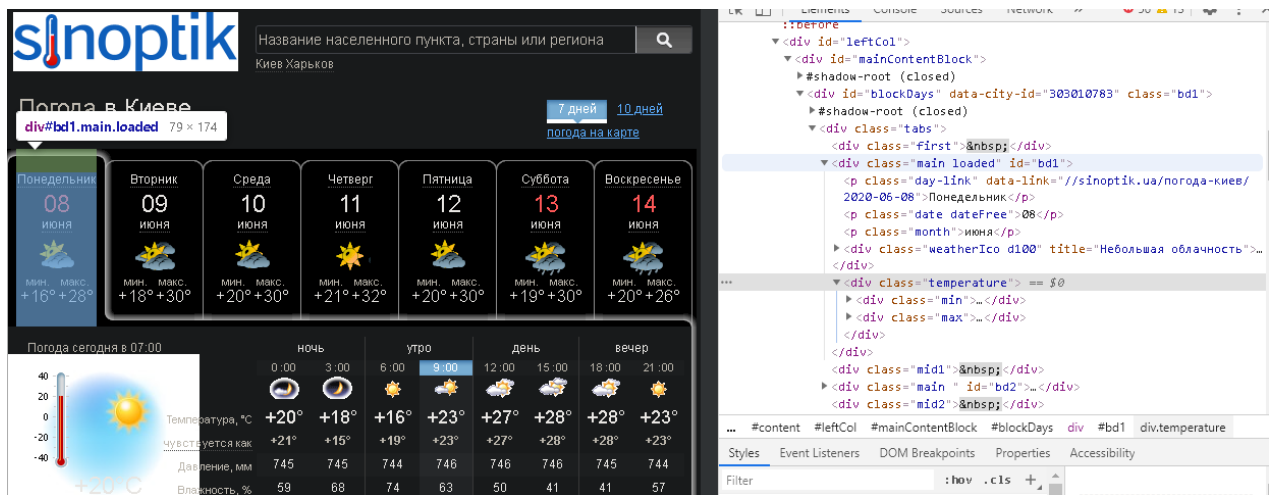


Рис. 4.10 Потрібний відділ з інформацією у програмному коді сайту

Знайшовши потрібні параметри витягуємо інформацію(рис 4.11).

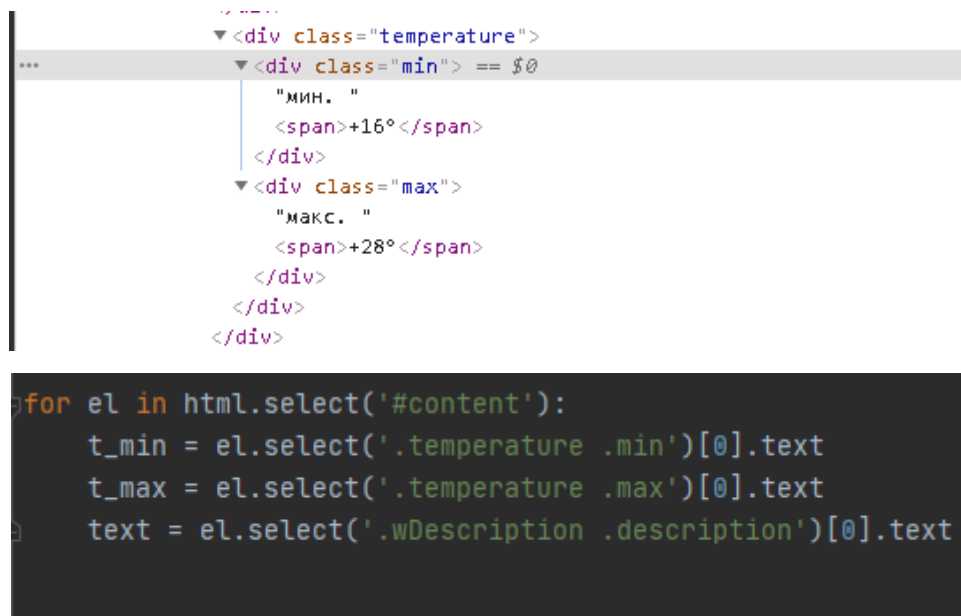


рис. 4.11 витягування інформації з програмного коду сайту

Створивши змінні, які у будь-який час будуть витягувати свіжу інформацію з сайту, можна виводити їх через телеграм-бота(рис. 4.12.):

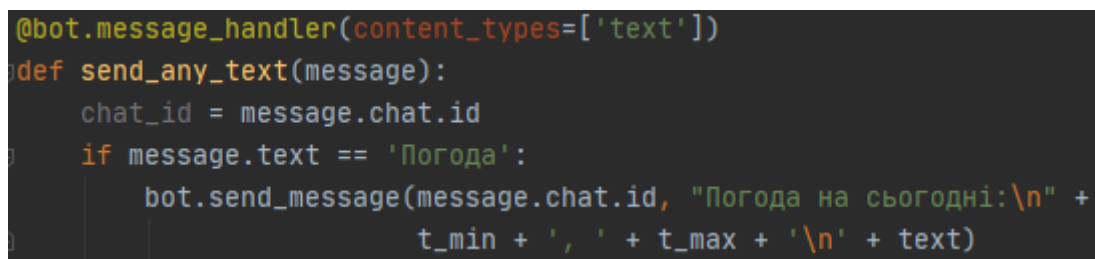


рис. 4.12 програмний код відправки інформації телеграм-ботом

Використовуючи тіж самі хендлери, але з параметром `content_types`, спонукаємо бота реагувати на потрібний розробнику тип даних – а саме текст, та виконувати вже певну потрібну дію, а саме відправка повідомлення з даними про погоду. Звіряємо інформацію(рис. 4.13):

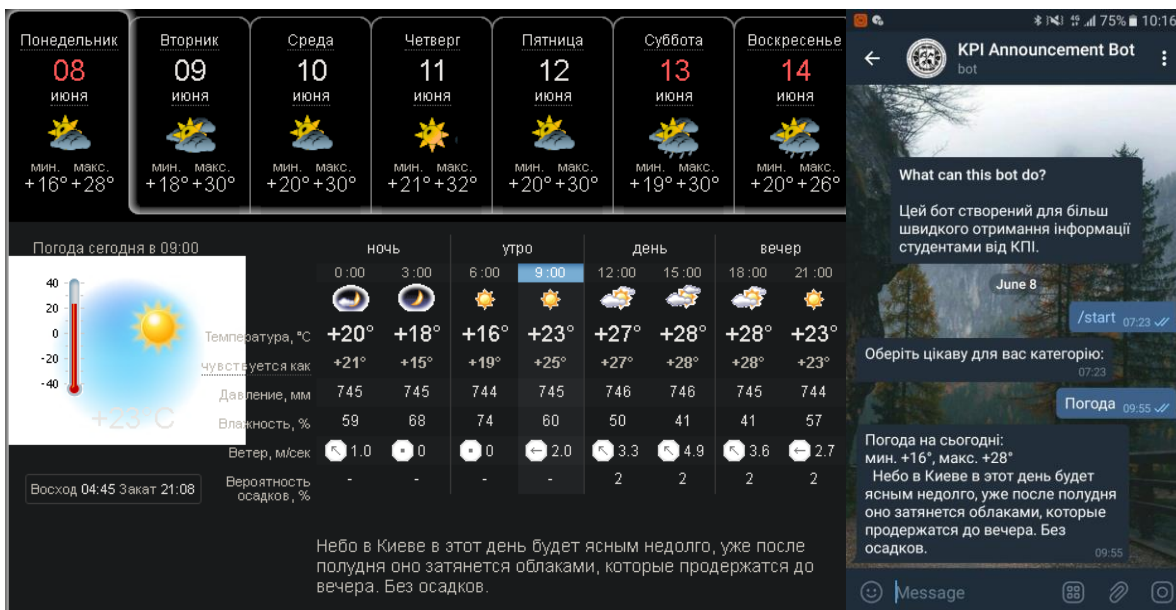


рис. 4.13 звірення отриманої інформації з сайту

Також телеграм-бот має функцію відправки будь-яких файлів користувачу, в даному випадку – методичних вказівок(рис. 4.14):

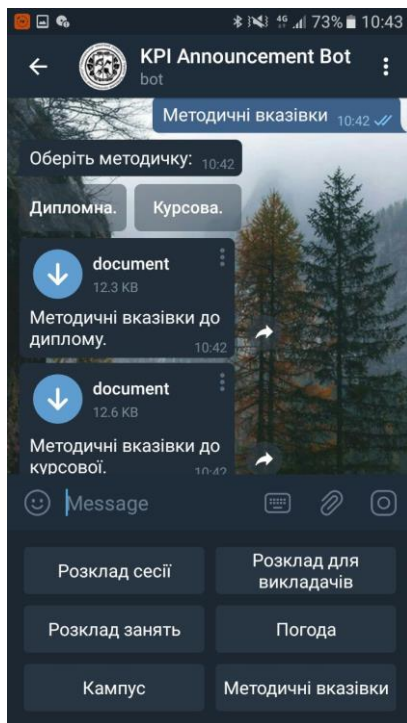


рис. 4.14 отримання методичних вказівок від телеграм-боту

Для початку було прийняте рішення для цього дипломного проекту – а саме створення інлайн клавіатури для варіацій у виборі користувача. Інлайн клавіатура – це така ж сама клавіатура з кнопками, але вона прикріплюється до повідомлення, завдяки цьому не заважаючи головній клавіатурі даного проекту(рис. 4.14).

Спочатку створюється сама інлайн клавіатура, яка прикріплена до повідомлення(рис. 4.15):

```
elif message.text == 'Методичні вказівки':
    markup = types.InlineKeyboardMarkup(row_width=2)
    item1 = types.InlineKeyboardButton("Дипломна.", callback_data='File1')
    item2 = types.InlineKeyboardButton("Курсова.", callback_data='File2')
    markup.add(item1, item2)
    bot.send_message(message.chat.id, 'Оберіть методичку:', reply_markup=markup)
```

рис. 4.15 створення інлайн клавіатури та її прикріплення до повідомлення

При цьому задається відповідний сигнал для кожною кнопки, завдяки параметру `callback_data='File1'` та `callback_data='File2'`, які в іншій функції будуть оброблюватися(рис. 4.16).

```
elif call.data == 'File1':
    with open(r"C:\Users\1\PycharmProjects\PyWorking\Files\Metod1.docx", "rb") as misc:
        f = misc.read()
        bot.send_document(chat_id=call.message.chat.id,
                           data=f,
                           caption='Методичні вказівки до диплому.')
elif call.data == 'File2':
    with open(r"C:\Users\1\PycharmProjects\PyWorking\Files\Metod2.docx", "rb") as misc:
        f = misc.read()
        bot.send_document(chat_id=call.message.chat.id,
                           data=f,
                           caption='Методичні вказівки до курсової.')
```

рис. 4.16 обробка на відповідні сигнали з кнопок та відправка документації

Виконується це в ще одному хендлері, який обробляє сигнали з кнопок(рис. 4.17):

```
@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
```

рис. 4.17 хендлер по обробці сигналів з різних кнопок

					ІАПЦ.467200.003 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступними трьома функціями цього телеграм-боту – є вивід розкладу занять, сесії та викладачів. Просто парсити сайт розкладу не вийде, тому що потрібно буде користувачу ввести найменування потрібної групи або викладача без помилок(рис. 4.18):

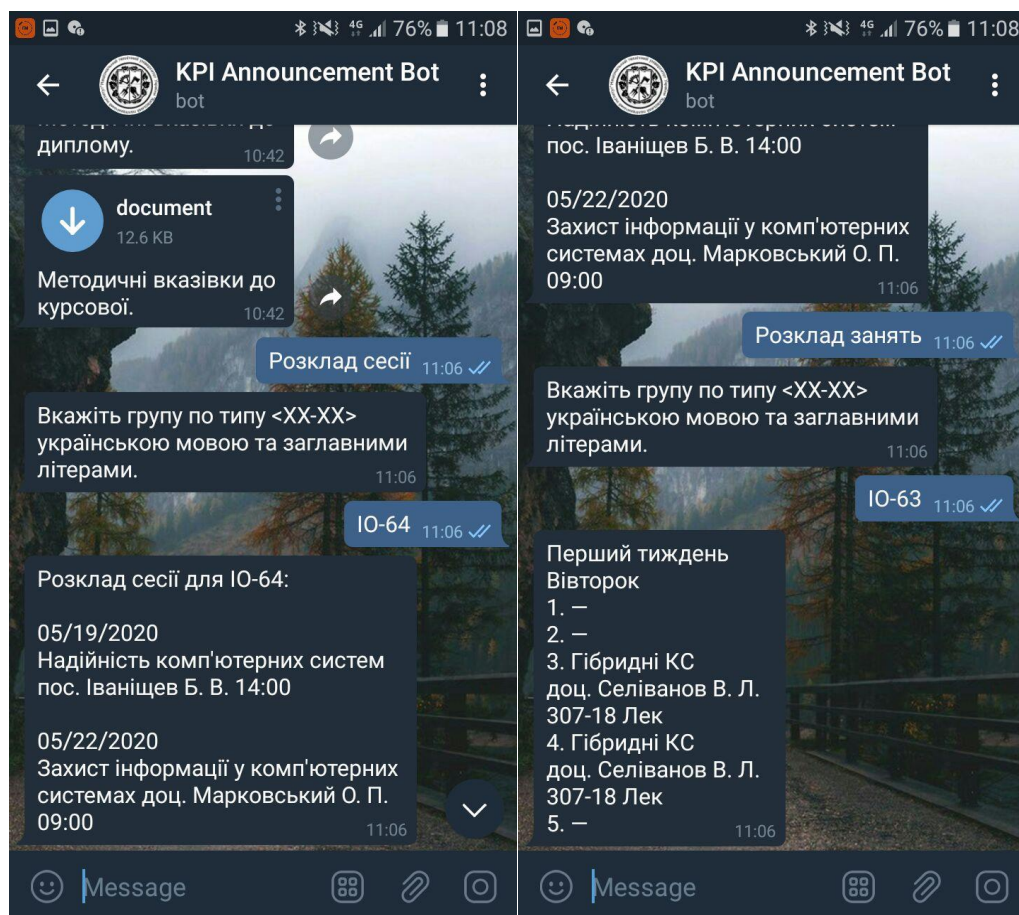


рис. 4.18 отримання розкладу сесії та занять для різних груп

Також перевіримо розклад для викладачів, заміняючи найменування групи на ПІБ викладача. Телеграм-бот записує отримані дані від користувача, вносячи їх в зміну та після цього вводить на сайті розкладу, витягуючи з отриманої сторінки потрібну інформацію(рис. 4.19).

Теж саме виконуємо з кампусом, але використовуємо вже дві змінні – для логіну та паролю.

При веденні невірної інформації – бот повідомить користувача про це, та буде очікувати вірної інформації(рис. 4.20).

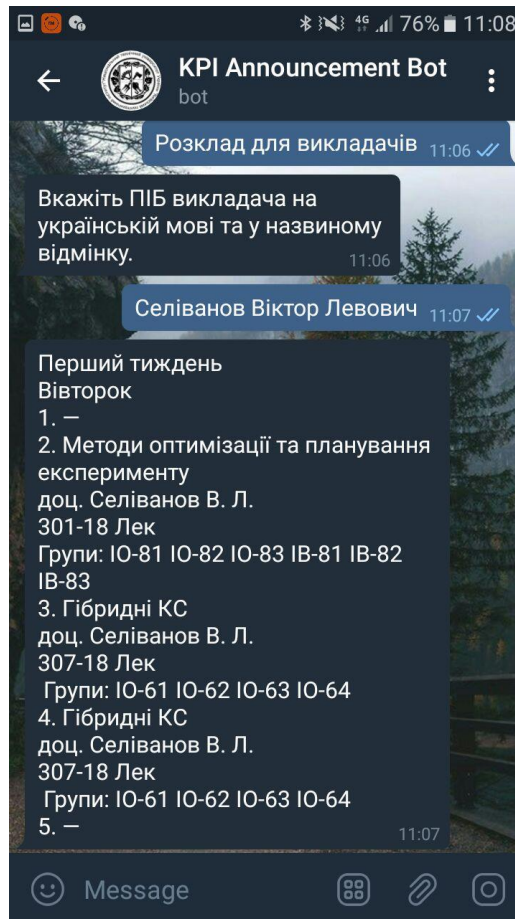


рис. 4.19 отримування розкладу для викладача

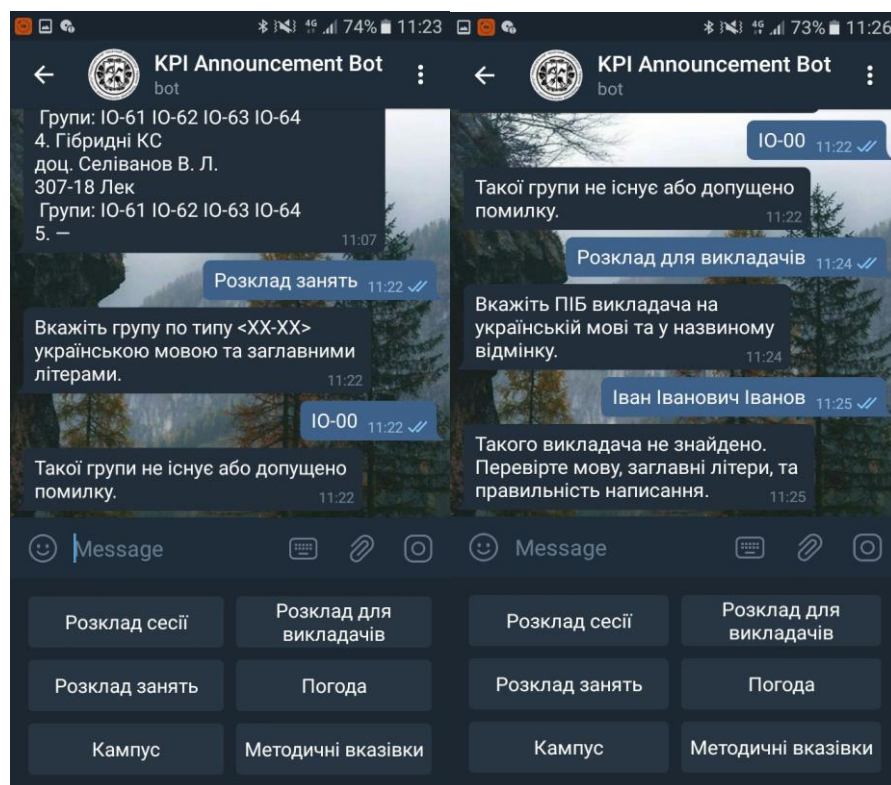


рис. 4.20 відправка повідомлення про помилку введення

					ІАПЦ.467200.003 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Також телеграм-бот використовує БД. Для цього дипломного проекту було обрано базу даних MongoDB.

Для початку треба зв'язати телеграм-бота з базою даних, використовуючи наступний код та бібліотеку pymongo(рис. 4.21):

```
connect_db = MongoClient('localhost', 27017)
db_schedule = connect_db[config.name_db]
logging.basicConfig(level=logging.INFO)
```

рис. 4.21 приєднання телеграм-бота до бази даних

У цьому дипломному проекті студенти та телеграм-бот будуть обмінюватися інформацією, при цьому бот буде запам'ятовувати студентів, перевіряючи їх chat_id.

Сама БД буде наповнена інформацією про групи університету та їх склад по системі(рис. 4.22):

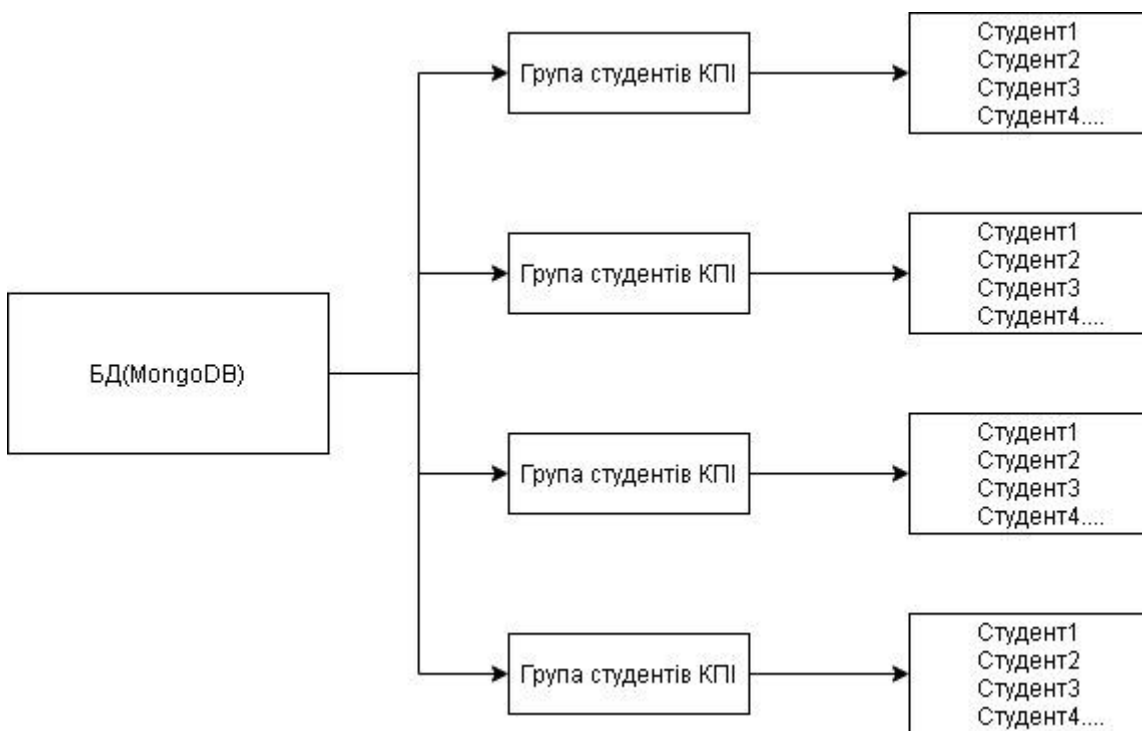


рис. 4.22 побудова БД для телеграм-боту

При цьому телеграм-бот буде доповнювати БД за допомогою запам'ятовування chat_id студентів, які вводили певну групу, тим самим прив'язуючи їх до групи(рис. 4. 23):

```

async def set_group(message: types.Message):

    try:
        _, group = message.text.lower().split(' ')
        groups = db_schedule.groups.find_one({ "group_full_name" : "{}".format(group)})['group_id']
        user = db_schedule.users
        user.update_one({'chat_id': '{}'.format(message.chat.id)}, {"$set": {'group': '{}'.format(groups)}})
        msg = "Вкажіть групу по типу <XX-XX> українською мовою та заглавними літерами."
        db_schedule.set_log.insert_one({'message': '{}'.format(message.text),
                                         'answer': 'True',
                                         'username': '{}'.format(message.from_user.username)})
        await message.reply(msg, reply_markup=create_main_markup())
    except:
        msg = "Такої групи не існує або допущено помилку."
        db_schedule.set_log.insert_one({'message': '{}'.format(message.text),
                                         'answer': 'False',
                                         'username': '{}'.format(message.from_user.username)})
        await message.reply(msg, reply_markup=create_main_markup())

```

рис. 4.23 додавання студенту у БД

При початку користування ботом, він буде перевіряти, чи внесений студент до БД(рис. 4.24):

```

user = db_schedule.users
if user.find_one({'chat_id': '{}'.format(message.chat.id)}) == None:
    user.insert_one({'chat_id': '{}'.format(message.chat.id),
                    'full_name': '{}'.format(message.from_user.full_name),
                    'username': '{}'.format(message.from_user.username)})
else:
    None

```

рис. 4.24 перевірка студента на наявність в БД

Отримав всю потрібну інформацію, бот виведе розклад по групі(рис. 4.25):

```

@dp.message_handler(regex='Розклад занять')
async def rozklad_zan(message: types.Message):
    group = db_schedule.users.find_one({ "chat_id" : "{}".format(message.chat.id)})['group']
    msg = func.rozklad_zan(group, func.day_number(), func.week_number())

```

рис. 4.25 функція виведення розкладу телеграм-ботом

ВИСНОВОК ДО РОЗДІЛУ 4

У ході цього розділу було наочно показано створення телеграм-бота. Також були описані усі умови створення бота – від отримання токену, до пояснення параметрів у програмному коді. Були розглянуті можливості бібліотеки Telegram Bot API та її наглядна реалізація в Python за допомогою певних бібліотек. Продемонстрована працездатність та справність бота, його функції та особливості.

					ІАПЦ.467200.003 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК ДО ДИПЛОМНОГО ПРОЕКТУ

На основі усієї виконаної роботи, були отриманні потрібні знання для створення застосунку для миттєвого оповіщення студентів – мети даного дипломного проекту.

Була детально розглянута сутність Telegram Bot API – основного інструменту у створенні ботів у Telegram. З'ясована розповсюдженість телеграм-ботів у сучасний час, їх функції, можливості та великого спрощення та зменшення рутинної праці звичайних людей. Телеграм-боти можуть виконувати розповсюджені дії, відповідати на часті запитання, та виконувати прості електронні дії. Була розглянута низка сучасних телеграм-ботів з доволі різними функціями: від компілятора програмного коду – до бота моніторингу погоди. Знайдені їх плюси та мінуси.

Оброблена велика кількість інформації про створення телеграм-ботів. Велика кількість мов програмування, які реалізують своїм функціоналом Telegram Bot API. Була обрана більш сприятлива мова програмування для проекту – Python та її середовище розробки - PyCharm IDE.

Був розглянутий інструментарій створення дипломного проекту - Python та PyCharm IDE. Знайдені їх сильні та слабкі сторони.

На практиці був створений з нуля телеграм-бот для дипломного проекту. Зроблен докладний опис застосунку для миттєвого оповіщення студентів з використанням великої кількості функцій, можливостей бібліотек та БД. Описані та продемонстровані можливості телеграм-боту, його швидкісну та якісну обробку інформації та передача її до користувача.

Основним недоліком даного дипломного проекту є його залежність від сайтів. А саме – від програмного коду сайтів. Телеграм-бот витягує інформацію з сайтів використовуючи певні кейси в програмному коді сайту, тому при зміні сайту – потрібно буде змінювати й програмний код телеграм-боту.

					ІАПЦ.467200.003 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Создаем Telegram бота на API.AI [Электронный ресурс] – 2017. – Режим доступа до ресурсу:
<https://habr.com/ru/post/336668/>.
2. Боты: информация для разработчиков [Электронный ресурс] – 2016. – Режим доступа до ресурсу:
<https://tlgrm.ru/docs/bots>
3. Telegram APIs [Электронный ресурс] – Режим доступа до ресурсу:
<https://core.telegram.org/api#bot-api>
4. Инструкция по созданию Telegram ботов. Часть 5. Пишем Telegram бота на php для работы через longpolling [Электронный ресурс]. – 2018. – Режим доступа до ресурсу:
https://radioham.ru/telegram_bot_5/
5. Python Telegram bot api. [Электронный ресурс] – Режим доступа до ресурсу:
<https://github.com/eternnoir/pyTelegramBotAPI/#types>
6. Примеры ботов [Электронный ресурс]. – 2017. – Режим доступа до ресурсу:
<https://tlgrm.ru/docs/bots/samples>
7. Примеры использования чат-ботов в бизнесе [Электронный ресурс]. – 2017. – Режим доступа до ресурсу:
<https://vc.ru/flood/25197-business-bot>
8. Rextester [Электронный ресурс] – Режим доступа до ресурсу:
https://ru.botostore.com/c/rextester_bot/
9. Бот поиска аномальных цен на авиабилеты [Электронный ресурс] – Режим доступа до ресурсу:
<https://bot.aviasales.ru/>
10. Создание бота в Telegram. Основы [Электронный ресурс] – Режим доступа до ресурсу:
<https://wibe.team/sozдание-bota-v-telegram/>
11. Serverless Telegram бот на базе PHP и AWS Lambda [Электронный ресурс]. –

					ІАПЦ.467200.003 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

2019. – Режим доступа до ресурсу: <https://devenenergy.ru/archives/941>
12. Веб-фреймворк Django (Python) [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django>
13. MySQL и MongoDB — когда и что лучше использовать [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://habr.com/ru/post/322532/>
14. Зонов К. Введение в MongoDB [Электронный ресурс] / Кирилл Зонов. – 2017. – Режим доступа до ресурсу: <https://mkdev.me/posts/vvedenie-v-mongodb>
15. Nuri H. Introduction to MongoDB [Электронный ресурс] / Halperin Nuri. – 2013. – Режим доступа до ресурсу: https://www.pluralsight.com/courses/mongodb-introduction?aid=7010a000002LUv2AAG&promo=&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=XYZ_EMEA_Dynamic&utm_content=&gclid=CjwKCAjwqpP2BRBTEiwAfpID-3LtLPnaT223CvUY8jybs4MNgtclMLq65pnRLM4BK06hPV1pbqRzoBoCKg8QAvD_BwE
16. MySQL [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://habr.com/ru/hub/mysql/>
17. TELEGRAM BOT WEBHOOK КАК НАСТРОИТЬ И ЗАПУСТИТЬ [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <http://be-miner.info/telegram-bot-webhook-kak-nastroit-i-zapustit/>
18. Telegram бот через webhook [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://retifrav.github.io/blog/2018/12/02/telegram-bot-webhook-ru/>
19. Welcome to Python Telegram Bot's documentation! [Электронный ресурс]. – 2015. – Режим доступа до ресурсу: <https://python-telegram-bot.readthedocs.io/en/stable/>
20. aiogram Documentation [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://readthedocs.org/projects/aiogram/downloads/pdf/latest/>
21. Работа с библиотекой pyTelegramBotAPI [Электронный ресурс]. – 2019. – Режим доступа до ресурсу:

https://groosha.gitbook.io/telegram-bot-lessons/extra/pytelegrambotapi_basics

22. Бот для Telegram на Python: от первой строчки кода до запуска на Heroku
[Электронный ресурс]. – 2017. – Режим доступа до ресурсу:

<https://tproger.ru/translations/telegram-bot-create-and-deploy/>

					ІАЛЦ.467200.003 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК 1

Застосунок для миттєвого оповіщення студентів

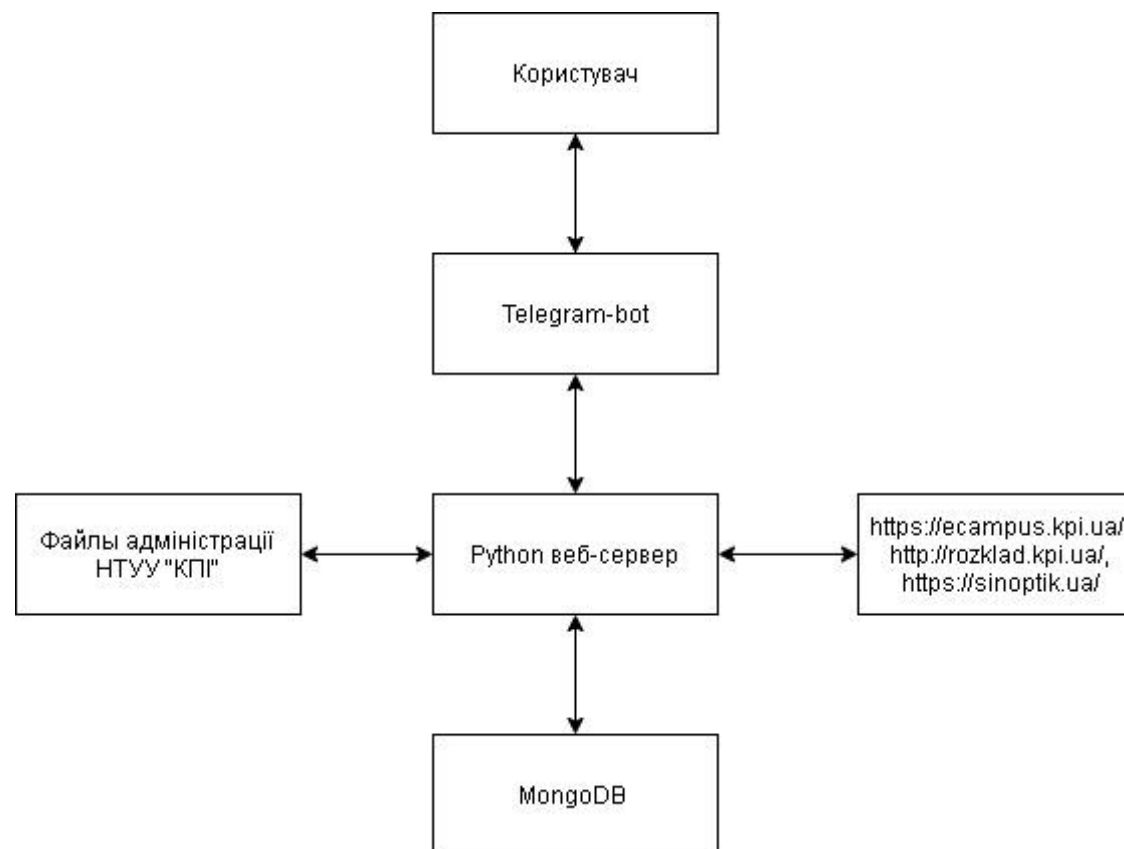
Застосунок для миттєвого оповіщення студентів

Схема структурна

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2020 р.



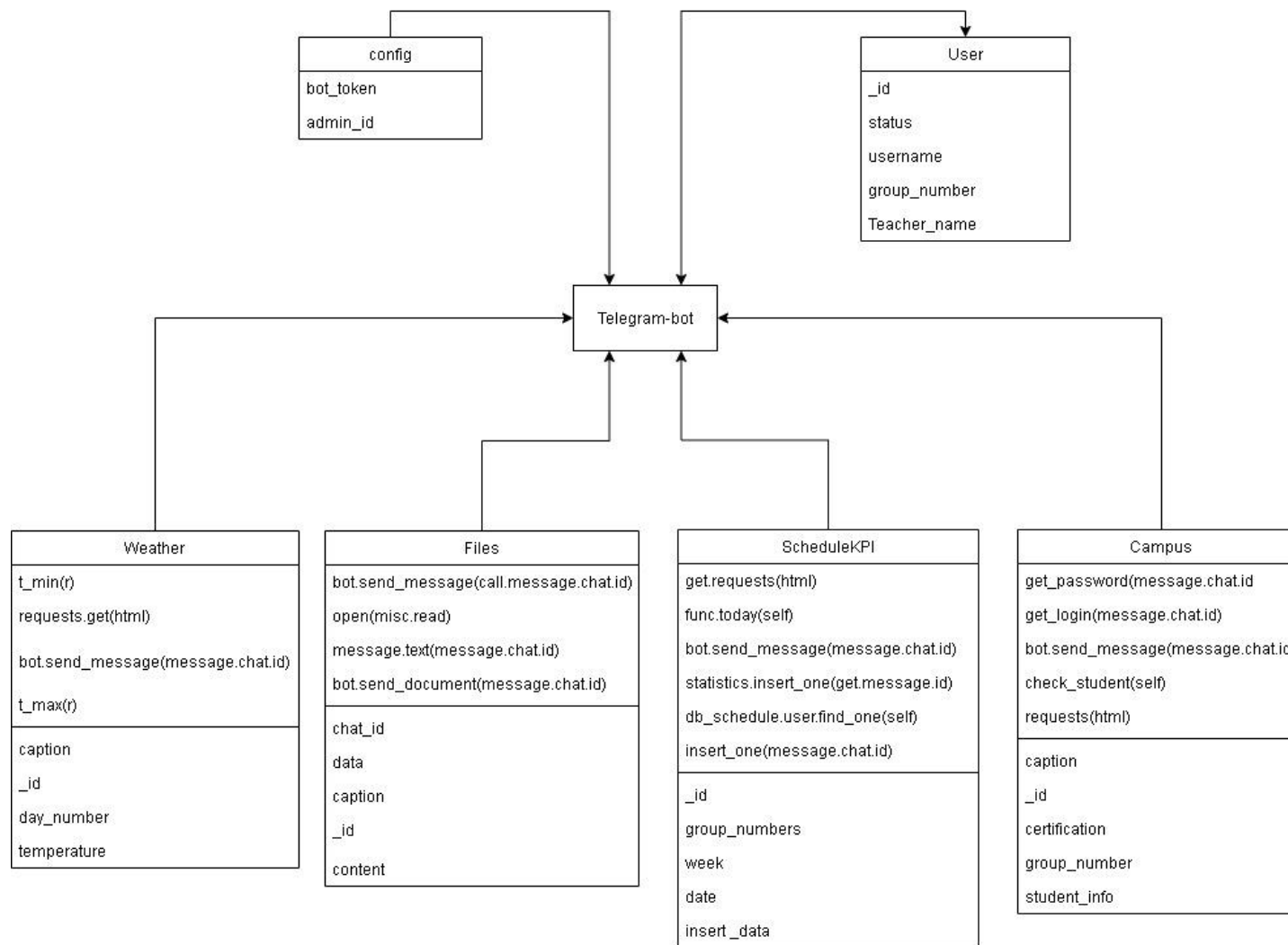
					ІАЛЦ.467200.004 Д1			
Зм.	Арк.	№ докум.	Підпис	Дата	Застосунок для миттєвого оповіщення студентів Схема апаратного забезпечення	Літ.	Аркуш	Аркушів
Розробив		Ткаченко П.О.					1	1
Перевірив		Габінет А.В.						
Реценз.								
Н. Контр.		Сімоненко В.П.						
Затв.		Стіренко С.Г.				НТУУ «КПІ», ФІОТ, ІО-64		

ДОДАТОК 2
Застосунок для миттєвого оповіщення студентів

Застосунок для миттєвого оповіщення студентів
Схема структурна
ІАЛЦ.467200.005 Д2

Аркушів 1

Київ 2020 р.



Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Ткаченко П.О.		
Перевірів		Габінет А.В.		
Реценз.				
Н. Контр.		Сімоненко В.П.		
Затв.		Стіренко С.Г.		

ІАЛЦ.467200.004 Д1

Застосунок для миттєвого оповіщення
студентів
Схема структурна

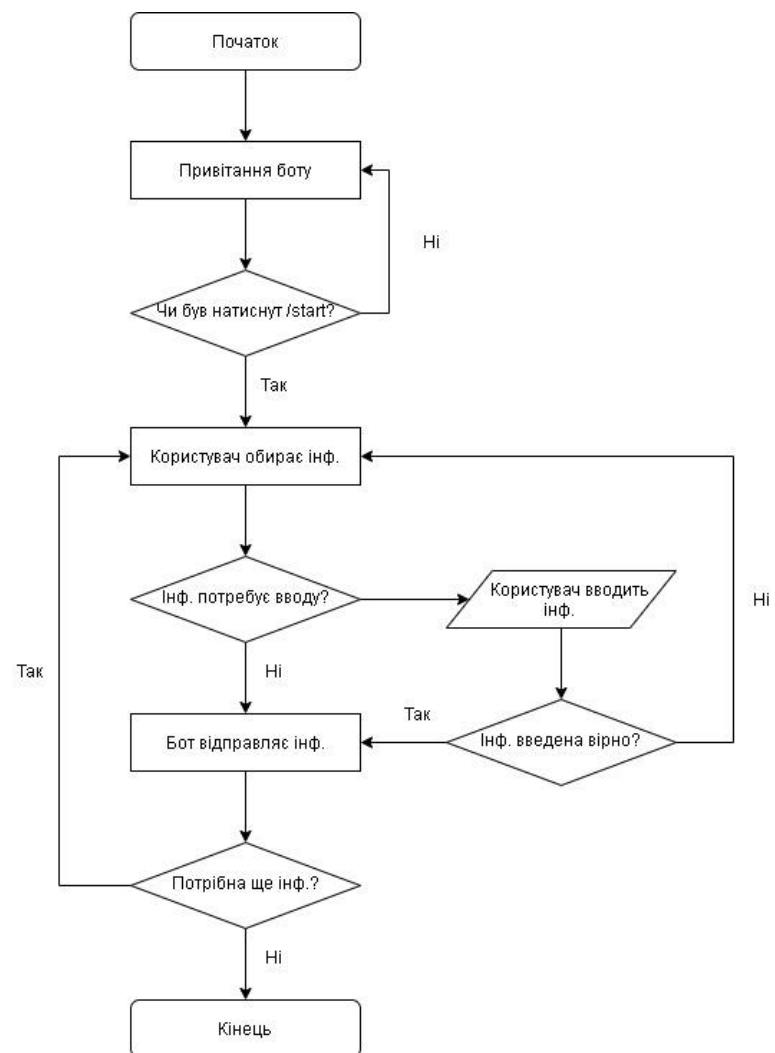
Лім.	Аркуш	Аркушів
	1	1
НТУУ «КПІ», ФІОТ, ІО-64		

ДОДАТОК 3
Застосунок для миттєвого оповіщення студентів

Застосунок для миттєвого оповіщення студентів
Схема функціональна
ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ 2020 р.



					ІАЛЦ.467200.006 ДЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Застосунок для миттєвого оповіщення студентів Схема функціональна			Лім.	Аркуш	Аркушів	
Розробив		Ткаченко П.О.									
Перевірив		Габінет А.В.							1	1	
Реценз.								НТУУ «КПІ», ФІОТ, ІО-64			
Н. Контр.		Сімоненко В.П.									
Затв.		Стіренко С.Г.									